

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH ŘÍDICÍHO SOFTWARE PRO RC SOUPRAVU

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Martin Faltíčko

BRNO 2012



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

NÁVRH ŘÍDICÍHO SOFTWARE PRO RC SOUPRAVU

IMPLEMENTATION OF SOFTWARE FOR RC RADIO

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Martin Faltíčko

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Zdeněk Vašíček

BRNO 2012

Abstrakt

Diplomová práce se zabývá vývojem řídicího software pro profesionální modelářskou rádiovou soupravu. Jejím výstupem je funkční řešení, které uspokojí požadavky stávajících leteckých i jiných modelářů a je konkurenceschopné vzhledem k ostatním značkám rádiových souprav. V práci je zahrnut návrh a implementace uživatelského rozhraní spolu s požadovanými datovými strukturami. Dále jsou diskutovány možnosti použití základních algoritmů pro řízení leteckých modelů.

Abstract

This document deals with the control software for professional Model-radio transmitter. The aim of the thesis is to design a functional solution that would satisfy the requirements of existing aircraft modelers (and others), thus to compete against other manufacturers of radio sets. In the following pages there is presented the design and implementation of user interface with regard to functionality and clarity, then the basic algorithms for driving aircraft models are explored. As a result of those findings the control software is implemented for an embedded device using the supplied hardware components.

Klíčová slova

RC souprava, Modelářský vysílač, technologie FHSS 2.4GHz, Mikrokontrolér STM32F405, ARM Cortex-M4, Bootloader.

Keywords

RC set, A transmitter for modellers, Technology FHSS 2.4GHz, Microcontroller STM32F405, ARM Cortex-M4, Bootloader.

Citace

Faltíčko Martin: Návrh řídicího software pro RC soupravu, semestrální projekt, Brno, FIT VUT v Brně, 2012

Návrh řídicího software pro RC soupravu

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Zdeňka Vašíčka. Další informace mi poskytli Ing. Stanislav Jelen (Jetimodel s.r.o.), Ing. Tomáš Klesnil (Jetimodel s.r.o.), Ing. Richard Šimeček (Jetimodel s.r.o.), Štěpán Škorpil (heliservis.eu) a Tomáš Ciniburk. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Jméno Příjmení (Martin Faltičko)
Datum (20. 5. 2012)

Poděkování

Chtěl bych poděkovat všem lidem z firmy Jetimodel s.r.o. za důvěru v tak rozsáhlý projekt, jakým profesionální rádiová RC souprava bezesporu je.

© Martin Faltičko, 2012

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů..

Obsah

Obsah.....	1
1 Úvod.....	3
2 Technologie v modelářství	4
2.1 Vývoj rádiového dálkového ovládání	4
2.2 Modelářské servo	5
2.3 Provedení dostupných RC souprav	6
3 Řízení modelů.....	9
3.1 Ovládání modelu letadla.....	10
3.2 Ovládání modelu vrtulníku	12
4 Analýza požadavků.....	14
5 Použitá hardwarová platforma	15
6 Implementace řídicího programu	18
6.1 Obsluha periférií	19
6.2 Základní funkčnost vysílače	20
6.2.1 Úkony po zapnutí	20
6.2.2 Transformační funkce.....	21
6.3 Navržené datové třídy	23
6.3.1 Třída String	23
6.3.2 Třída Vector	23
6.3.3 Třída JSON	23
6.3.4 Třída Locale.....	25
6.4 Přehrávání audia	25
6.5 Připojení k USB	26
6.6 Grafické uživatelské rozhraní	26
6.6.1 Základní formulářové komponenty	27
6.6.2 Struktura menu	29
6.6.3 Hlavní obrazovka	30
6.6.4 Základní nastavení modelu	31
6.6.5 Nastavení pro letové režimy	33
6.6.6 Další volby vztažené k modelu	36
6.6.7 Možnosti přehrávání audio souborů, alarmy a časovače	37
7 Shrnutí.....	41
7.1 Metriky	42
8 Závěr	43

9	Použité zkratky a termíny.....	44
	Dodatek A – Nastavení vývojového prostředí a ovládání aplikace emulátoru	46
	Dodatek B – Struktura doprovodného CD.....	47

1 Úvod

Tato diplomová práce se věnuje vývoji moderního RC vysílače, a to především z pohledu softwarové stránky. Cílem je vytvořit funkční produkt, který splní základní požadavky kladené současnými modeláři s ohledem na funkčnost i ergonomii.

V práci je věnován prostor především pilotům modelů letadel a vrtulníků, neboť tyto dvě skupiny mají nejvíce specifické potřeby pro řízení svých strojů. Ostatní modeláři (např. „autíčkáři“ nebo „lodičkáři“) mají daleko jednodušší požadavky, takže se většinou přizpůsobí možnostem vysílače.

Specifiky různých modelářských oborů se zabýval již semestrální projekt, který této práci předcházel. V něm byl vysvětlen základní způsob řízení modelů a také byly představeny běžně dostupné technologie používané v modelářství. Diplomová práce jej rozšiřuje především o implementační část a vyhodnocení.

Práce je členěna následovně. V první kapitole se čtenář seznámí se stručnou historií modelářství s ohledem na vývoj RC souprav a použitých technologií. Následně jsou představeny vybrané elektronické komponenty použité při řízení modelů, s jejich výhodami i nevýhodami.

Druhá kapitola je věnována základním poznatkům z teorie pilotáže a jsou probrána specifika jednotlivých typů modelů. Další kapitoly sumarizují tato specifika a na základě nich je vytvořen seznam konkrétních požadavků na nový vysílač. Je také představena použitá hardwarová platforma a následně je detailně popsán implementovaný software i s ilustracemi nejzajímavějších nabídek.

Závěrečná část se zabývá vyhodnocením dosažených výsledků a diskutuje možnosti dalšího vývoje.

2 Technologie v modelářství

V modelářství jsou odjakživa uplatňovány nové technologie, přičemž jsou častokrát nalezeny alternativní způsoby jejich použití anebo jsou přínosy nových poznatků vytěženy na maximum. Jako příklad můžeme uvést modelářskou závodní kategorii F5B, kde jsou plně využity možnosti moderních nabíjecích baterií LiPol a jsou také uplatněny nejnovější poznatky z aerodynamiky [13].

2.1 Vývoj rádiového dálkového ovládání

Různé pokusy o dálkové rádiové řízení objektů je možné datovat až do doby druhé světové války. V té době byl asi nejznámější projekt Afrodita [2], v němž byly použity upravené bombardéry B-17, z nichž byly odstraněny standardní doplňky jako kulometná výzbroj a místa pro posádku. Namísto nich byl letoun vybaven dvěma kamerami s elektronikou pro bezdrátové řízení a napěchován několika tunami výbušnin. Ovládání probíhalo ze sekundárního letounu CQ-17. Takto měly být ničeny německé bunkry a jiné opevněné cíle, avšak metoda dálkového řízení se ukázala v té době nespolehlivá a nebezpečná. Proto byl tento projekt také ukončen.

Po válce probíhaly amatérské experimenty s elektronickým řízením modelů (podrobně popsáno v [14]). Samotná elektronika ale byla velmi hmotná, vyžadovala speciální bateriový zdroj žhavícího napětí a umožňovala řízení pouze jedné funkce na modelu a to ještě skokově. Elektronky také odebíraly nemalé množství proudu.

V padesátých letech byly vyrobeny tranzistory, což znamenalo první etapu „miniaturizace“ v modelářství. Elektroniku si stále stavěli amatérští elektrotechnici, profesionální výroba přišla o několik let později. Stále šlo o jednopovelové soupravy, později ale přišla další vylepšení v podobě krystalem řízených superhetů, které přinesly zlevnění technologie a lepší VF selektivitu a spolehlivost. Následně se objevily vícekanálové soupravy a ovládání již bylo plně proporcionální (tzn., bylo umožněno plynulého pohybu řídicích ploch, např. pomocí servomechanizmů). Modulace byla nejprve amplitudová (známé pásmo 27MHz, které se nyní používá snad jen pro hračky), pak přišla frekvenční (na pásmech 35MHz pro letecké modeláře a 40MHz pro ostatní).

Digitální soupravy, v nichž dochází po přijetí dat z řídicích pák k jistému stupni zpracování, přišly v devadesátých letech. Ty již umožňovaly v podstatě takové možnosti nastavení, jaké nabízí současné soupravy. Jmenovitě jde např. o vysílače Graupner MC-24 nebo Futaba FC-28. Ty se mezi modeláři vyskytují v hojném počtu i dnes.

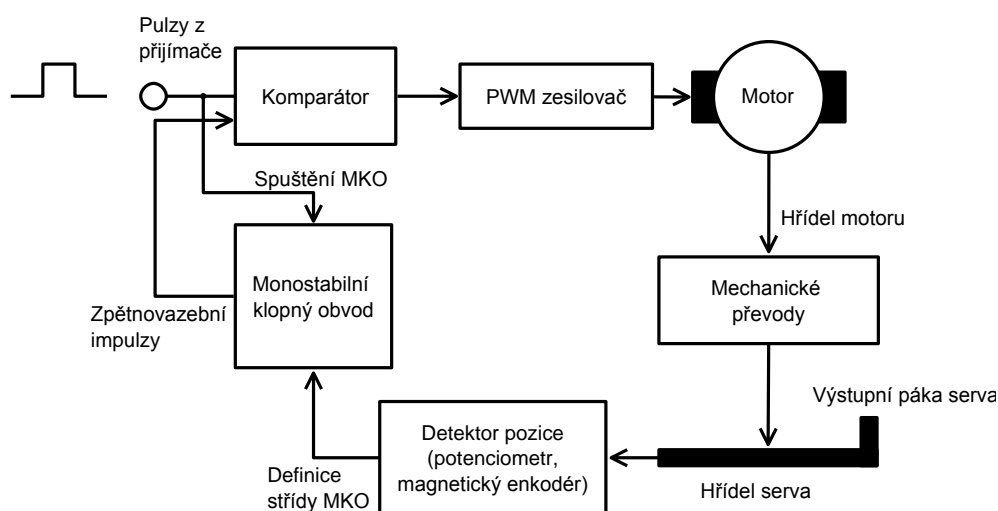
Mezi lety 2009 a 2010 začali výrobci souprav postupně přecházet na bezlicenční pásmo 2,4GHz, což umožnilo nadále zvýšit stupeň miniaturizace (již není zapotřebí rozměrná anténa) a také

spolehlivosti – díky použitému kódování *FHSS* [12] je podstatně omezena možnost rušení a není zapotřebí komplikovaně zjišťovat, který kanál je volný.

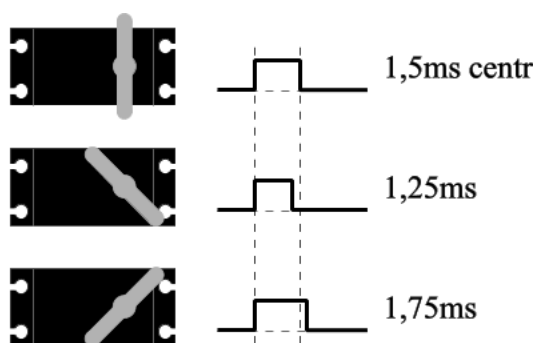
Nyní se na trhu objevuje stále více produktů nabízejících telemetrii, tedy přenos dat z modelu v reálném čase. Navíc vzrůstá potřeba mít naměřená telemetrická data kde uložit pro pozdější zpracování a vyhodnocení.

2.2 Modelářské servo

Modelářské servo nebo zkráceně též „servo“ je označení pro elektromechanickou komponentu, která umožňuje měnit seřízení pohyblivých ploch modelu, čímž operátorovi poskytuje kontrolu nad objektem. Blokové schéma analogového serva je znázorněno na Obrázku 1. Servo je připojeno pomocí třížilového vodiče k přijímači a na vstupu očekává PWM signál o šířce 1-2ms a periodě přibližně 20ms. Doba, kdy je signál aktivní, určuje, jakou silou a do jaké pozice se servo má natočit. Tento efekt je patrný z Obrázku 2.



Obrázek 1: Schematické znázornění struktury analogového serva (převzato z [5]).

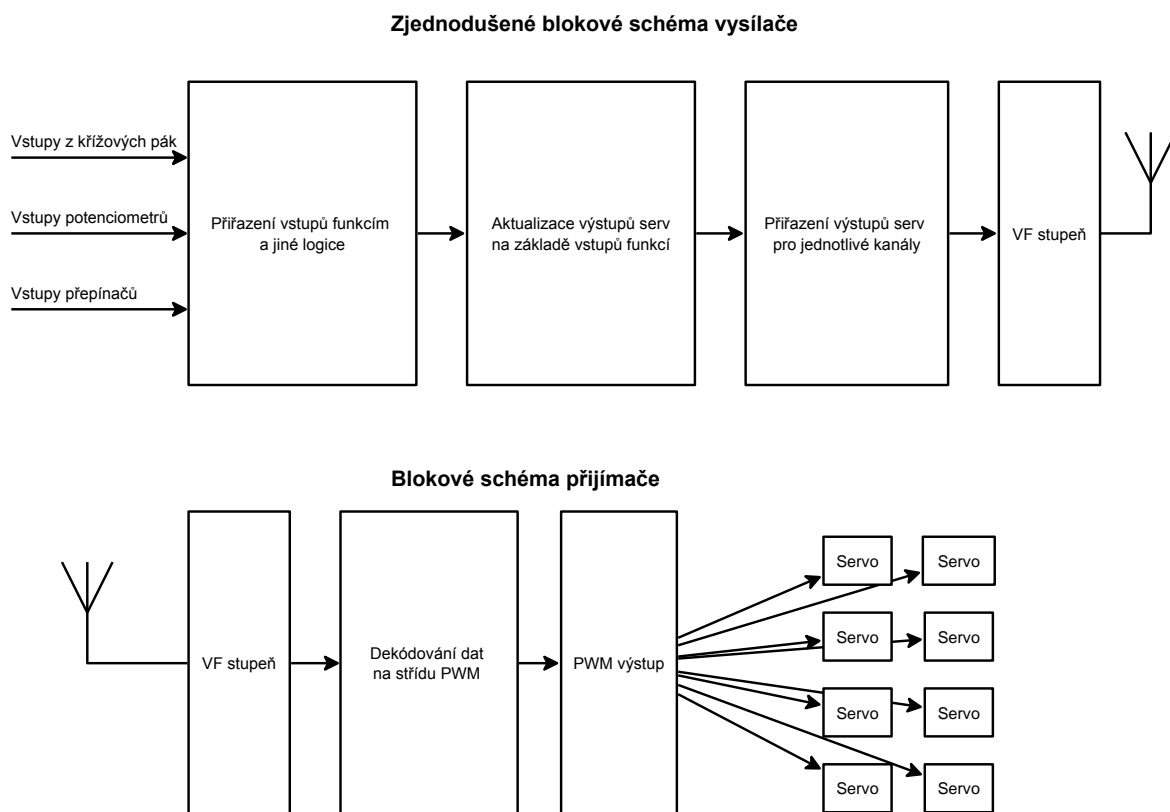


Obrázek 2: Úhel natočení serva v závislosti na délce signálu (převzato z [4]).

Serva mají bohužel jednu nevýhodu. Jsou to jednoduché komponenty bez jakékoli možnosti nastavování, již je velmi často potřeba. Častokrát není například možné do modelu instalovat servo mechanicky tak, aby byla využita jeho plná úhlová dráha nebo naopak, aby svým pohybem nepřekračovalo mechanické limity ovládané plochy a nenávratně ji tak poškodilo. Z tohoto důvodu byly vyvinuty procesorové vysílače, které kompenzují mechanická omezení serva a umožňují pohodlné přenastavení parametrů jako střed serva, výchylku nahoru a dolů nebo dokonce i zpomalení rychlosti přeběhu z jedné krajní pozice do druhé.

2.3 Provedení dostupných RC souprav

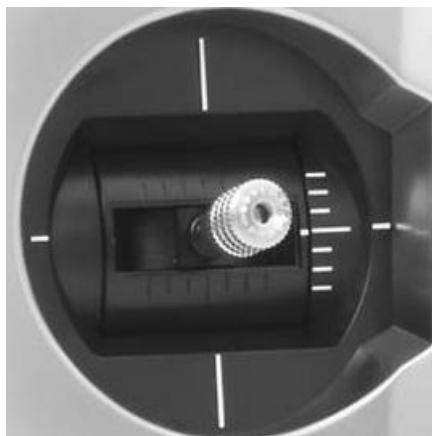
Na Obrázku 3 je uveden zjednodušený průběh řízení RC modelu – od detekování pozic ovladačů, zpracování těchto dat a vyslání přes rádiový modul – až po přijetí signálu přijímačem a jeho následnou distribuci pro jednotlivá serva.



Obrázek 3: Zjednodušené schéma RC vysílače a přijímače.

K detekci pozic ovladačů slouží většinou přesné potenciometry (u proporcionálních pák) nebo binární vstupy procesoru v případě přepínačů. Poloha potenciometrů je snímána pomocí rychlých AD převodníků – nyní standardně dvanáctibitových. V nejnovějších aparaturách se již objevují i Hallové sondy namísto potenciometrů. Jsou totiž přesnější, detekce je bezdotyková a sondy tedy mají téměř neomezenou životnost.

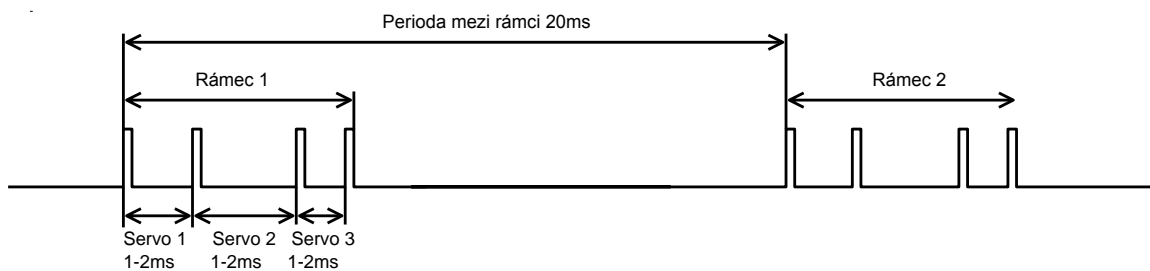
Standardní křížový ovladač, jakým jsou vybaveny téměř všechny modelářské vysílače, je zobrazen na Obrázku 4. Tento ovladač v sobě obsahuje dva potenciometry (pro každou osu jeden) a uchycení os bývá uloženo v kuličkových ložiscích.



Obrázek 4: Křížový dvouosý ovladač RC soupravy Futaba T8FG.

Vstupy ovladačů jsou zpracovávány hlavním řídicím procesorem. V jednoduchosti můžeme říci, že transformuje informace o pozici pák na výstupy pro jednotlivé kanály a provádí korekce pro každý výstupní kanál zvlášť. Procesor musí běžet na real-time bázi, neboť je životně důležité dodržet maximální časové limity pro odezvu systému na změnu v řízení. Obvyklá maximální doba odezvy leží mezi 20 a 30ms, moderní rychlé soupravy pracující v pásmu 2,4GHz však mohou mít tuto dobu i poloviční.

Vysokofrekvenční modul zajišťuje jednosměrný přenos řízení z vysílače k přijímači. Mnoho let převládalo kódování **PPM** (Pulse Position Modulation, použito pro AM i FM v pásmech 27 až 40MHz). Při něm jsou jednotlivé kanály vysílače seřazeny podle pořadí na přijímači a do éteru je vždy vyslán ve své podstatě digitální rámec, kdy hodnota log. 1 znamená „oddělovač kanálů“. V tomto rámci je pozice serva zakódována pomocí časového intervalu mezi dvěma oddělovači, kdy je signál v log. 0. Časové intervaly pro každé servo trvají standardně 1-2ms (1ms – levá krajní výchylka, 2ms – pravá krajní výchylka) a opakují se přibližně 50krát za sekundu. Je možné vypočítat, že více než 10 kanálů tento systém není schopen ovládat. Jednoduše proto, že by hrozilo riziko vzájemného překrývání rámců mezi sebou. Na Obrázku 5 je možné si prohlédnout časovou posloupnost této modulace.



Obrázek 5: Časová posloupnost PPM modulace.

Druhou alternativou přenosu informace z vysílače do modelu je **PCM** kódování (Pulse Code Modulation) – datový přenos paketů. Zde jsou využity standardní postupy přenosu digitální informace, včetně kontroly integrity paketů pomocí jednoduchého CRC. Tato technika umožňuje ovládání více než deseti funkcí a je přesnější než PPM. Také je možné dosáhnout nižších latencí. Nevýhodou je, že když dojde k rušení, zahazují se celé pakety a model se stává neovladatelným. U PPM by v té době byly alespoň některé funkce částečně řiditelné.

Dříve byly modulací PCM vybavovány pouze ty nejlepší vysílače, avšak nyní při použití pásma 2,4GHz je již některá podoba PCM nutností. Bohužel ale každý výrobce používá jiný formát přenosu, a tak jsou produkty mezi sebou vzájemně nekompatibilní.

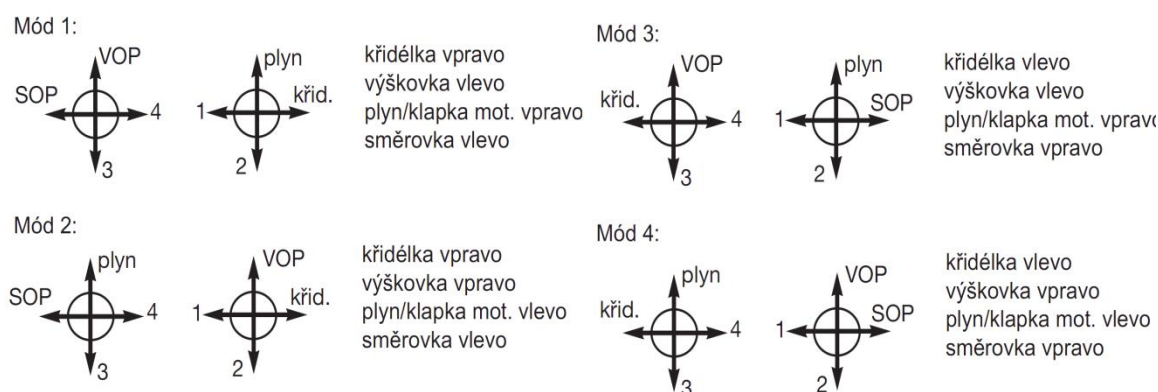
3 Řízení modelů

Řízení modelů je poměrně komplexní záležitost, neboť každý model má svoje specifika a navíc se mohou vzájemně lišit návyky jednotlivých pilotů. Naštěstí však existují určitá nepsaná pravidla, jichž se většina výrobců drží – takže pro českého modeláře není příliš velký problém používat japonský (Futaba), německý (Graupner) či americký (Spektrum) vysílač.

V zásadě platí rozdělení vysílačů na „palcové“ a „pultové“ – podle způsobu držení při řízení. Palcové typy drží člověk v dlaních, přičemž pohybuje ovládacími pákami pouze prostřednictvím palců. Pultové jsou naopak celé zasazeny do opěrného pultu, který je zavěšen na pilotově ramenním popruhu. Dlaně se potom celou vahou opírají o pult a pilot může páky vysílače svírat např. mezi palcem a ukazovákem.

Základem každého vysílače je dvojice dvouosých křížových ovladačů (Obrázek 4 zachycuje jeden z nich). Každá osa ovladače je brána jako vstup pro konkrétní funkci modelu. Podle toho o jakou funkci se jedná, je možné rozdělit vysílače do několika módů (viz Obrázek 6), z nichž nejobvyklejší jsou módy 1 a 2. U prvního je páka plynu vpravo současně s křídélky, kdežto v druhém módu je ovladač plynu na opačné straně. Jelikož jsou všechny ovladače vycentrované pružinkami a pouze plyn je mechanicky aretován a bez centrování, je většinou při změně módu také nutný zásah do vysílače. Moderní soupravy ale takovou změnu umožňují, a tak si modeláři nemusejí pořizovat speciální verzi, která by jim v tomto jedině vyhovovala. Namísto toho koupí univerzální provedení, jež si posléze sami upraví.

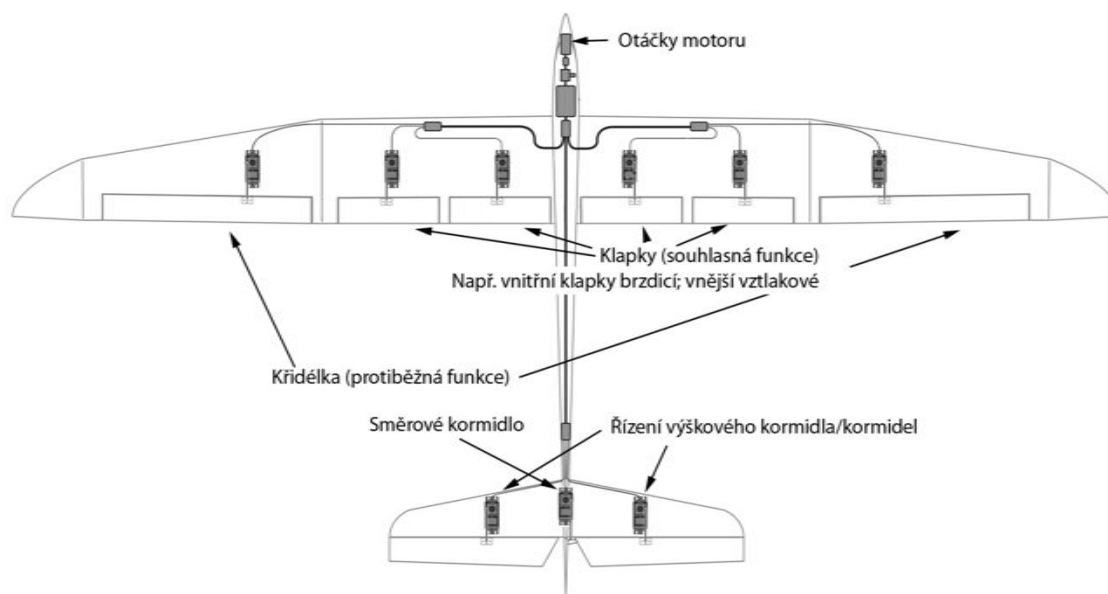
V následujících kapitolách jsou shrnuty základní termíny použité v oblasti pilotáže modelů letadel a vrtulníků. Podrobnější výklad je možné nalézt v [15], z něhož bylo čerpáno.



Obrázek 6: Schematické znázornění základních módů křížových ovladačů. Převzato z [9].

3.1 Ovládání modelu letadla

Jak již bylo výše zmíněno, pohyblivé plochy modelu letadla jsou ovládány pomocí modelářských serv. Jimi je možno řídit např. velikost výchylky směrového či výškového kormidla, křidélek anebo brzdících štítů (viz Obrázek 7). Otáčky motoru jsou regulovány buď samostatným servem (spalovací motory) nebo elektronickým regulátorem otáček. Ten je pak připojen jako klasické servo a dostává od přijímače totožné impulzy.



Obrázek 7: Funkce ovládané v modelu letadla. Podle [11].

Častokrát se stává – a neplatí to jen při záletu nového modelu – že letadlo neletí v přímém směru tak, jak by se dalo očekávat. Je potřeba jej tzv. **dotrimovat**, což znamená, že se jemně upraví středové pozice serv tak, aby byl let přímočarý. V současnosti k trimování slouží rotační enkodéry, vždy jeden pro každou osu, nebo čtveřice tlačítek umístěných v blízkosti křížových ovladačů.

Je také potřeba poznamenat, že nejen skutečná letadla, ale i modely prochází během letu několika etapami. Jedná se o tzv. **letové fáze** (popř. **letové režimy**), při nichž je možné nezávisle nakonfigurovat některé funkce modelu. Například start a přistání u maket je možné prohlásit za zvláštní letový režim, při kterém se vysouvá podvozek a spouští vztlakové klapky. Pokud by teoreticky bylo možné vysunout podvozek v jiném letovém režimu, hrozilo by nebezpečí, že jej pilot spustí omylem, případně dojde i k poškození modelu. Jako další letové režimy můžeme označit i konfigurace větroně pro různé úkony. Při kroužení v termice mírně vysouvá vztlakové klapky tak, aby měl co nejmenší klesavost a nejlépe využil stoupavé proudy. V dalším stádiu, když termika slábne nebo se pilot rozhodne letět jinam, konfigurace se přenastaví na „rychlost“ a může být bez problémů proveden přeskok skrze oblasti s horšími podmínkami.

Některé funkce modelu letadla jsou dále specifické. Pokud vezmeme v úvahu křidélka – těm je obvykle nastavena určitá míra **diferenciace**, kdy se křidélko pohybuje směrem nahoru s větší výchyl-

kou než dolů. Diferenciace slouží především k odstranění nežádoucího chování modelu a to především u modelů s nesouměrným (prohnutým) a tenkým profilem křídla. Změnou výchylky křídélka totiž vzroste u nesouměrného profilu aerodynamický odpor rozdílně na každé straně křídla a jedna polovina je tímto aerodynamickým odporem bržděna více než druhá. Model po vychýlení křidélek nejdříve mírně vybočí na stranu, kde je křídélko vychýleno dolů, a teprve potom se začne otáčet kolem své podélné osy.

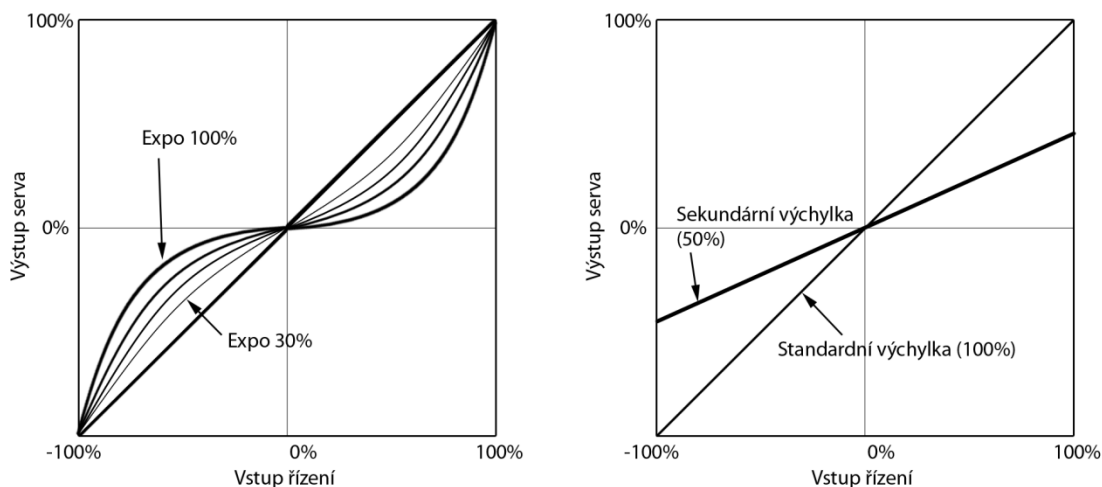
Někdy také modeláři potřebují nastavit dvě různé míry výchylek pro jednotlivé kanály. Tzv. **dvojí výchylky** se použijí v případě, kdy jsou žádané razantní reakce při provádění akrobacie, avšak je potřeba zachovat vlačné odezvy při vzletu a přistání. Přepínání mezi těmito mírami výchylek je pak prováděno pomocí přepínače.

S dvojími výchylkami také souvisí možnost nastavení **exponenciálního průběhu** odezvy serva. Někdy model např. reaguje velmi citlivě na drobné změny v řízení, takže je často hůře ovladatelný. Exponenciální průběh zaručí, že odezva serva při pohybu páky okolo jejího středu bude výsledná reakce serva menší, než když se páka bude pohybovat u krajních pozic. Exponenciální průběh se často nastavuje v součinnosti s dvojími výchylkami. To proto, aby bylo docíleno konstantní odezvy při pohybu páky okolo středu s tím, že krajní výchylky budou různé. Více napoví Obrázek 8.

Jelikož jsou spotřební vysílače vybavovány nevýkonnými procesory a výpočet exponenciální funkce bývá výpočetně náročný, bývá exponenciální průběh nahrazen křivkou polynomu třetího stupně, jež je téměř shodná. V následujících rovnicích (1) a (2) jsou uvedeny vztahy výpočtu aproximace exponenciálního průběhu pro FPU i celočíselnou aritmetiku. Proměnné k , l představují míru vlivu kubické křivky, $k \in \langle 0,1 \rangle$, $l \in \langle 0, ExpoLimit \rangle$. *ExpoLimit* a *xMax* jsou vhodně zvolené celočíselné konstanty.

$$y_{fpu} = k \cdot x^3 + (1 - k) \cdot x \quad (1)$$

$$y_{int} = \frac{l \cdot x^3}{xMax^2} + \frac{(ExpoLimit - l) \cdot x}{ExpoLimit} \quad (2)$$

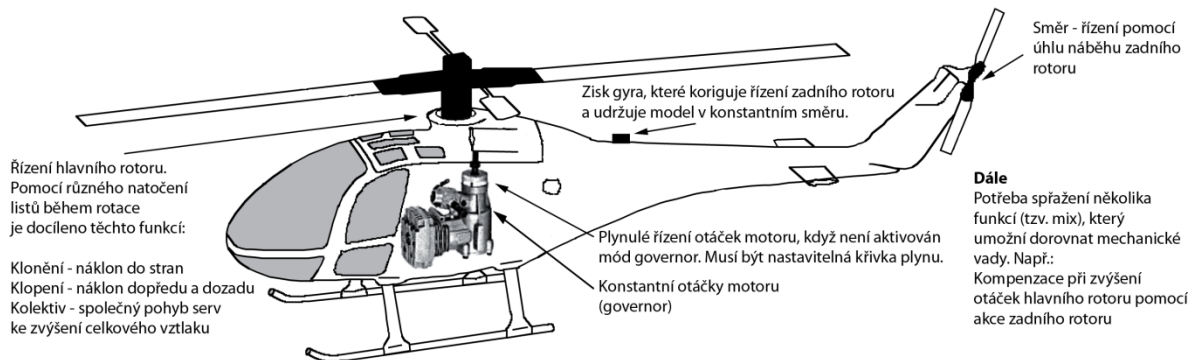


Obrázek 8: Exponenciální průběh dráhy serva a dvoji výchylky.

3.2 Ovládání modelu vrtulníku

Vrtulník je přirozeně nestabilní systém a jako takový jej pilot musí mít neustále pod kontrolou. Moderní elektronika, jako jsou **modelářská gyra**, sice dovolují potřebu rychlých reakcí potlačit, avšak vrtulníky mají stále řadu zvláštních aspektů. Na Obrázku 9 jsou zobrazeny potřebné funkce k jeho ovládání.

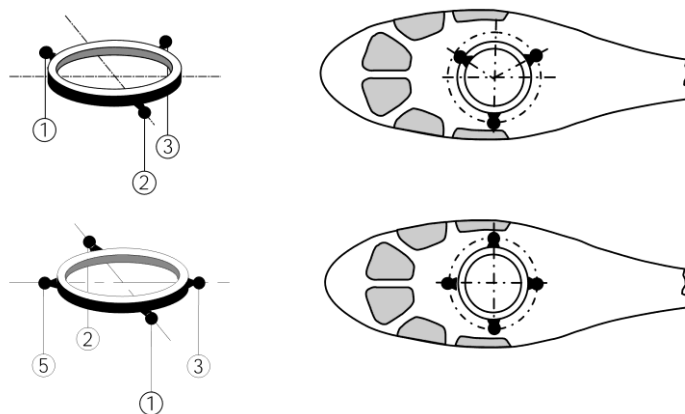
Nejprve zmiňme modelářské gyro. To v současné době obsahuje typicky MEMS gyroskop, který umožňuje stabilizaci vrtulníku podél vertikální osy. Připojuje se mezi přijímač a vyrovnávací servo ocasního rotoru, takže úpravou signálu z přijímače pomáhá v rychlé stabilizaci směru modelu. U každého gyra je možné nastavit zisk, což je míra, v jaké gyro zasahuje do výstupu z přijímače. Zisk se nastavuje elektronicky pomocí dalšího kanálu. Pokud je příliš velký, dochází při stabilizaci k překmitávání a řízení je nepohodlné. Naopak při malém zisku nedochází k ustálení vůbec.



Obrázek 9: Funkce, které je nutné ovládat u modelu vrtulníku. Vytvořeno podle [10].

Rotory modelů vrtulníků jsou, stejně jako jejich předlohy, řízeny pomocí pohyblivé **desky cyklicky** (angl. swashplate). Na této desce kloužou hlavy rotorových listů a kopírují její polohu, což se projeví ve změně úhlu náběhu listů. Takto je možné docílit stavu, kdy má rotor vrtulníku v každém bodě jiný vztlak a je tedy dostupný pohyb vpřed i do stran.

Způsob řízení desky cyklicky může být jednoduchý – to v případě, že máme zkonstruován mechanický mix, který pro každý kanál (představující jednu osu) nastaví desku do požadované polohy. V každém jiném případě je nutné použít softwarové mixování jednotlivých kanálů tak, aby výstupem byla přímo pozice řídicích serv. Jejich nejčastější rozmístění je zobrazeno na Obrázku 10. Horní náčrt znázorňuje uchycení desky cyklicky pomocí tří serv. Mezi různými typy vrtulníků existují varianty tohoto uchycení, které se vzájemně liší úhlem, jaký serva mezi sebou svírají a jak jsou natočena vůči podélné ose trupu. Na spodním náčrtu je zobrazeno čtyřservové uchycení, které je používáno především u velkých maket.



Obrázek 10: Standardní schéma rotorové cyklicky (převzato z [10]).

Při ovládání vrtulníku, který má některou variantu uchycení desky cyklicky z Obrázku 10, jsou aplikovány následující postupy:

Kolektivní řízení – nastavení úhlu náběhu listů rotoru se mění kolektivně, tedy o stejnou výchylku a ve kterékoliv poloze. Tato funkce se využívá pro stoupání a klesání.

Cyklické řízení – nastavení úhlu náběhu listů rotoru je prováděno cyklicky – rotorový list má v každém místě během otáčky jiný úhel náběhu. Dochází tak ke klonění (naklání na strany, obdobně jako fungují křídélka u letadel) či klopení (naklápění vpřed a vzad; odpovídající funkcí u letadla je výškovka).

U vrtulníku je také nutné kvalitně regulovat otáčky motoru tak, aby byly ideálně konstantní při různých režimech letu. Tohoto je dosaženo několika způsoby. Jednak je nastavována vícebodová uživatelská křivka průběhu plynu v reakci na pozici ovladače, dále je možné spřahovat některé funkce, aby pracovaly v součinnosti se změnou plynu. Jako další alternativa se jeví speciální elektronická komponenta, tzv. **governor**, jenž udržuje konstantní otáčky motoru zcela automaticky.

4 Analýza požadavků

V následujícím seznamu uvedme základní požadavky na vyvíjený vysílač z pohledu softwarové stránky tak, aby splňoval potřeby uživatelů. Podle těchto požadavků se bude řídit pozdější grafický návrh jednotlivých položek v menu, ale i implementace konkrétních ovládacích subsystémů.

- Dostatečný počet ovládaných funkcí je základem. Fyzických kanálů je celkem šestnáct (omezení stávající technologie bezdrátového přenosu firmy Jetimodel [16]), takže je možné mít i šestnáct funkcí při mapování 1:1. Standardní funkce z kapitoly o řízení modelů jsou také zahrnuty.
- Každé funkci by mělo být možné přiřadit libovolný ovladač, potenciometr nebo přepínač jako hlavní vstup.
- Výstupy funkcí pro serva by mělo být možné libovolně přeuspořádávat a každé servo by mělo být samostatně kalibrovatelné. Zde se jedná o nastavení středu, výchylek a zpomalení rychlosti přejezdu.
- Mělo by být možné volit mezi různými typy modelů. V úvahu připadá rozdělení na obecné modely (pozemní technika, lodě), vrtulníky a plošníky (u nich je ještě možné dělení na akrobatické modely a větroně, jejichž řízení se někdy liší).
- U funkcí by mělo být možné nastavit exponenciální průběh, dvojí výchylky a také aplikovat uživatelské vstupně-výstupní křivky.
- Možnost libovolného přiřazení mixů mezi funkcemi.

Další požadavky se týkají funkčnosti časovačů a telemetrie. Časovače jsou důležité pro soutěžní piloty, kteří potřebují přesně změřit dobu letu, ale i pro ty, kteří chtějí předejít vybití pohonné baterie tím, že si nastaví předem známý čas motoru jako výchozí hodnotu pro odpočet. Přenos telemetrie zase zvyšuje informovanost pilota o událostech odehrávajících se v modelu a tím i zvyšuje bezpečnost provozu. Takže požadavky jsou následující:

- Mělo by být možné vytvořit několik časovačů, které budou operovat nezávisle.
- U každého časovače by měl být přiřazen spínač, jímž by byl spouštěn a zastavován.
- Měla by být implementována možnost zvukové signalizace při odpočtu.
- Pro soutěžní piloty je rovněž důležitá funkce mezičasů.
- Telemetrie by měla fungovat pro současná čidla systému Duplex EX a mělo by docházet k zobrazování zvolených údajů v reálném čase na vysílači.
- Měla by být přítomna možnost aktivování zvukového alarmu při překročení krajních hodnot měření. Např. pokud teplota motoru přesáhne limitní mez, nebo dojde k nadměrnému vybití baterií v modelu, je akustické varování nedocenitelné.
- S akustickým varováním souvisí i možnost přehrávání zvukových záznamů, ideálně v některém komprimovaném formátu jako MP3 či WMA, avšak postačí i WAV.

5 Použitá hardwarová platforma

Na Obrázku 11 je znázorněno blokové schéma vysílače DC-16, navrženého a zkonstruovaného firmou Jetimodel s. r. o. Hotový prototyp je pro ilustraci na Obrázku 12. Jako hlavní procesor byl zvolen řadič firmy ST Microelectronics, jelikož ta nabízí solidní SW podporu pro své produkty a obchodní zastoupení má i v ČR.

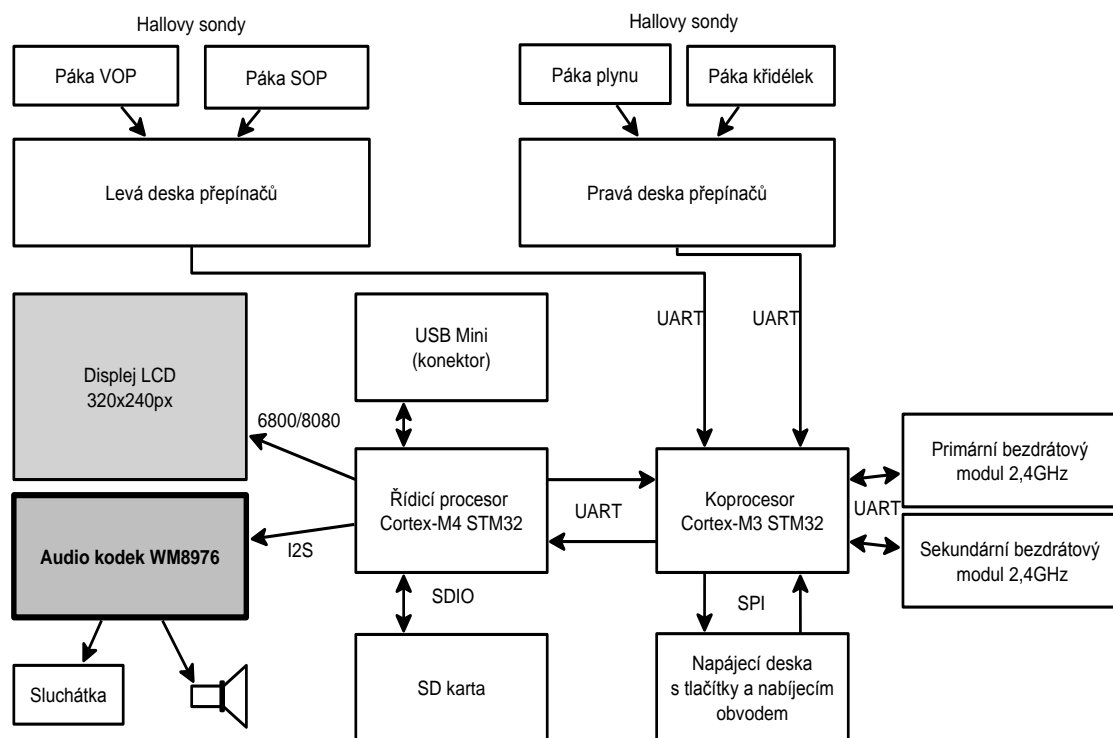
Konkrétně byl použit 32bitový procesor STM32F405RGT6, který běží na jádru ARM Cortex-M4 s frekvencí až 168MHz, je bohatě vybaven periferiemi a obsahuje dostatek paměťového prostoru. Parametry procesoru jsou následující:

- 1 MB interní paměti Flash, vestavěná jednotka FPU se single precision.
- 196 KB paměti SRAM.
- Podpora externí statické RAM nebo NAND Flash paměti připojené pomocí rozhraní FSMC.
- Napájecí napětí 2,0 až 3,6V.
- Možnost připojit krystalový oscilátor 4-25MHz. Interní obvod PLL z něj odvodí takt jádra.
- Vestavěný obvod RTC.
- Tříkanálový dvanáctibitový AD převodník.
- Několik 16bitových časovačů s možností IC/OC/generování PWM.
- Komunikační rozhraní USB 2.0 full speed, SDIO, I²S, SPI, I²C a USART.
- Řadič DMA.

Hlavní procesor obstarává veškeré životně důležité výpočty, stará se ale také o méně podstatné služby, jako je vykreslování GUI přes připojený monochromatický LCD displej s rozlišením 320×240px. Druhý procesor – rovněž STM32 – provádí předzpracování dat z přepínačových desek a křížových ovladačů. Ovladače jsou vyrobeny s ohledem na co nejvyšší životnost, a proto jsou také osazeny moderními bezdotykovými Hallovými sondami. Ty umožňují měřit přesnou intenzitu stacionárního magnetického pole s rozsahem až 4096 hodnot a v intervalu několika málo milisekund. Jako výstup potom poskytují PWM signál s určitou střídou.

Vlastní bezdrátový přenos probíhá přes proprietární modul Jeti Duplex, který pracuje s obousměrným přenosem v pásmu 2,4GHz. Je možné vysílání i současný příjem dat z modelu. Modul pracuje na principu FHSS, kdy dochází mnohokrát za sekundu ke vzájemnému přeladění obou komplementárních modulů.

Poslední komponentou systému je audio kodek, zde použitý WM8976 od firmy Wolfson. Jedná se o vysoce kvalitní Hi-Fi DA převodník (98dB SNR), který nabízí přímý výstup na sluchátka i pro reproduktor s impedancí 8Ω. Je řízen přes sběrnici I²C a samotná data jsou posílána po synchronní lince I²S, vždy v konfiguraci dva bajty na audiokanal.



Obrázek 11: Blokové schéma uvedené platformy. Popisovaná aplikace byla implementována pro řídicí procesor Cortex-M4.

Jak je vidět z Obrázku 11, koprocetor Cortex-M3 (typu STM32F103) maximálně ulehčuje hlavnímu mikrokontroléru práci s obsluhou sběrnic. Právě kvůli zjednodušení komunikace jsou oba propojeny pouze rozhraním UART – koprocetor poskytuje data z periférií a jako odpověď očekává pouze jednoduchou zprávu s výstupními hodnotami kanálů. Řadiče nejsou spolu vůbec nijak časově svázány, nikdy se nečeká na potvrzení o úspěšném doručení zprávy. Data jsou však v pravidelných intervalech přeposílána znovu, takže vliv na bezpečnost provozu je zanedbatelný.

Koprocetor obsluhuje levou a pravou desku přepínačů, přičemž v každé je zabudován osmibitový mikrokontrolér. Přepínače umístěné na těchto deskách jsou vyměnitelné a vždy obsahují zvláštní RC člunek. Což znamená, že podle doby nabíjení kondenzátoru je možné určit typ přepínače a pak také s ním správně zacházet. V současnosti je možné rozlišit, jestli se jedná o dvoupolohový či třípolohový přepínač nebo jednopolohové tlačítko. Informace o pozicích přepínačů se pravidelně posílají koprocetoru společně s hodnotami naměřenými Hallovými sondami v hlavních pákách.

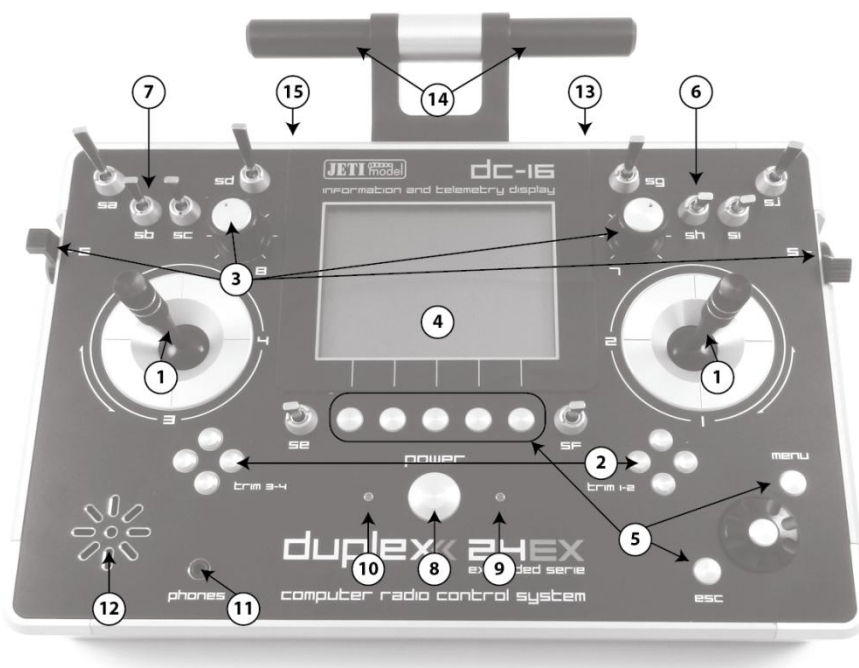
Údaje o pozicích přepínačů jsou v koprocetoru spojeny do jedné zprávy dohromady se stavy tlačítek a rotačního enkodéru, jež byly získány od mikrokontroléru umístěného v desce napájení. Celá zpráva je pak přeměrována do hlavního řadiče. Tímto je dosaženo co nejmenší potřeby komunikace mezi částmi systému a je tak možné získat relativně nízké latence.

Hlavní procesor, jenž obsahuje kompletně načtenou datovou strukturu modelu, získané vstupní informace vyhodnotí, přetransformuje je na kanálové výstupy a pošle zpátky koprocetoru. Poté jeho grafický subsystém provede detekci stisku tlačítka a v případě úspěchu zareaguje daným způsobem.

Jakmile koprocesor dostane paket s výstupy jednotlivých kanálů, přepoše je již jako hotový datový rámec do vysokofrekvenčních modulů. Vysílací moduly (VF) jsou dva z důvodu zvýšení bezpečnosti a mohou pracovat v několika společných režimech. V základním režimu vždy fungují oba moduly současně s jedním přijímačem, avšak data jsou posílána ve střídavém intervalu tak, aby se zaručilo, že alespoň jeden modul a jeho anténa budou mít nějaký signál.

Ve druhém režimu, tzv. *TwoWays*, vysílací moduly komunikují nezávisle na sobě se dvěma různými přijímači. Ty mohou být spojené přes inteligentní slučovač (např. JETI Enlink), případně jsou základní funkce řízení rozděleny mezi oba přijímače. Jedna část modelu je pak ovládaná přes jeden přijímač z prvního vysílacího modulu a zbytek modelu je řízen druhým přijímačem/modulem. Zálohován je vždy případný výpadek jednoho z přijímačů nebo vysílacího modulu.

Třetí režim VF modulů se nazývá *Učitel/Žák*. Sekundární vysílací modul je v něm vyhrazen pro komunikaci mezi dvěma vysílači, s modelem pak komunikuje vždy pouze vysílač učitele. Je-li DC-16 v režimu „Učitel“, primární VF modul komunikuje s modelem a ten sekundární přijímá od žakovského vysílače údaje o pozicích ovladačů. Při tomto režimu musí být vždy jeden vysílač označen jako učitel (master) a druhý jako žák a oba musí být navzájem spárované.



Obrázek 12: Ilustrační foto vysílače. Prvky: 1) levý a pravý pákový ovladač, 2) tlačítkové kříže trimů pákových ovladačů, 3) volně přiřaditelné proporcionální ovladače, 4) grafický displej, 5) navigační tlačítka a rotační enkodér, 6) levá (ze strany plošných spojů) deska s přepínači, 7) pravá deska s přepínači, 8) tlačítko *Power*, 9) zelená LED (*zapnuto*), 10) červená LED (*nabíjení*), 11) zdířka pro sluchátka, 12) reproduktor, 13) konektor USB Mini, 14) zdvojená anténa a současně úchop, 15) konektor napájení.

Na Obrázku 12 je vyfotografována čelní strana vysílače s popisky. Nejmarkantnějšími prvky na fotografii jsou dvouosé pákové ovladače vyrobené z kovu v kombinaci hliník/ocel. Jejich uložení je realizováno pomocí pěti kuličkových ložisek.

Celá konstrukce vysílače je velmi robustní. Cílem designu bylo dosáhnout maximální možné životnosti při provozování v poměrech polních podmínek. Tomu pak na druhou stranu odpovídá i větší hmotnost oproti ostatním vysílačům a požadavky na kvalitu se nepříznivě projeví také na výsledné ceně.

6 Implementace řídicího programu

Celý systém se chová jako real-time aplikace rozšířená o uživatelské rozhraní. Je zde nutné dosáhnout dodržení různých časových limitů, které jsou dány několika fakty, mimo jiné např. potřebou plnit periferie zvukového rozhraní daty v dostatečném předstihu. Musí tedy být vytvořeno několik samostatných úloh s různými prioritami, které budou běžet pod real-time operačním systémem. Pro výběr vhodného operačního systému se vyskytlo několik kandidátů, z nichž hlavními byli ti:

μC/OSII (www.micrium.com). Výborně zdokumentovaný systém propagovaný komerční firmou Micrium. Obsahuje velké množství knihoven pro dodatečnou práci se souborovým systémem nebo i TCP/IP stackem. Bohužel jeho zdrojové kódy jsou sice vydané a volně ke stažení, avšak licence pro komerční použití vyžaduje platbu vysokých vstupních částek, které začínají na 1000USD. Za další SW moduly do systému je také nutno dodatečně platit.

FreeRTOS (www.freertos.org). Open-Source systém vyvíjený Richardem Barrym je otevřený OS se základní funkcionalitou (úlohy, synchronizace, mutexy), je bez jakýchkoli licenčních poplatků a existuje pro mnoho platforem. Prvotní testovací verze aplikace běžely na tomto systému, avšak vyskytly se problémy se synchronizováním mezi přerušeními a úlohami a také s dodržováním časových limitů.

CoOS [6]. Tento real-time systém je svým API podobný jako FreeRTOS, rovněž zdarma, svým zaměřením se však specializuje pouze na mikrokontroléry Cortex-M3 a je pro ně optimalizován. Poskytuje nejlepší dobu odezvy v porovnání s jinými OS a při jeho použití v podstatě stačí nahradit volání funkcí FreeRTOS jejich ekvivalenty u CoOS.

Vzhledem ke svým pozitivním vlastnostem, jednoduchosti, rychlosti a nenáročnosti na paměťové místo byl nakonec pro výslednou aplikaci vybrán CoOS. K němu bylo pouze nutné doprogramovat podporu pro jádra Cortex-M4 s FPU jednotkou, neboť oficiální distribuce ji zatím nenabízí. Implementace této podpory spočívala v přidání assemblerových rutin ukládání a obnovování kontextu registrů FPU jednotky.

6.1 Obsluha periférií

Kromě operačního systému je neméně důležité zvolit vhodnou programovou podporu pro daný hardware. Jelikož firma Jetimodel s. r. o. vybrala jako hlavní řídicí jednotku procesory STM32 [8], je hojně používáno SW komponent od jejich výrobce, ST Microelectronics. Jedná se především o následující nízkourovňové ovladače pro přístup k perifériím.

SDIO Driver slouží pro čtení a zápis dat z připojené SD karty. SD karta slouží k trvalému uchovávání konfigurací modelů po vypnutí vysílače, ale jsou na ní uložena i další data jako zvukové soubory a telemetrické záznamy jednotlivých letů.

TIM komponenta je použita pro řízení časovačů a čítačů, které jsou využity pro přesné měření času, jehož granularitu je možné měřit až na úrovni milisekund.

DMA komponenta slouží pro rychlé přenosy dat mezi pamětí a jinými perifériemi. Umožní znatelné zrychlení určitých částí programu, neboť není nutné starat se o ruční přenos dat mezi sběrnici (ať již pomocí „data polling“ nebo přerušení). Data jsou přenesena automaticky a transparentně.

Použity jsou dále ovladače pro sběrnice **SPI** (Serial Peripheral Interface), **UART** (Universal Asynchronous Receiver/Transmitter), **I²C** (Inter-Integrated Circuit) a **I²S** (Integrated Interchip Sound). Tyto sběrnice slouží pro komunikaci mikrokontroléru s ostatními obvody, mezi něž patří jiné mikrokontroléry, čidla nebo zvukový kodek/zesilovač.

Jelikož je ve vysílači instalován konektor mini USB, byl použit i **ovladač sběrnice USB** (slave device). V současné době je USB již standardním portem v téměř všech elektronických zařízeních. U vysílače je použito zejména pro přístup na SD kartu skrze PC.

Poslední důležitou externí SW knihovnou je ta pro práci se souborovým systémem SD karty. Na ní bývá v naprosté většině instalován systém FAT (12/16/32) a pro přístup k datům se tedy jeví výhodně některá knihovna pro práci s tímto systémem. Z dostupných volných řešení vyšla nejlépe knihovna **FatFS** [7], jejíž API se velmi podobá standardnímu rozhraní jazyka C pro PC. FatFS obsahuje mnoho funkcí pro čtení i zápis do souborů a je relativně rychlá, dosahuje až 20MB/s při kontinuálním čtení za použití rozhraní SDIO a plného taktu sběrnice 25MHz. Je určena pro vestavěná zařízení, takže má i nízký tzv. „memory footprint“. Knihovna je volitelně rozšiřitelná o podporu dlouhých jmen u FAT32 a také umožňuje práci ve vícevláknovém prostředí.

6.2 Základní funkčnost vysílače

Uvnitř hlavního procesoru běží čtyři nezávislé úlohy operačního systému CoOS. První se jmenuje *DisplayTask* a jejím úkolem je primárně obsluhovat události spojené s vykreslováním obsahu displeje a řešit kompletní uživatelské rozhraní i s detekcí stavu tlačítek.

Druhá úloha, *AudioTask*, je součástí zvukového subsystému a byla vytvořena z toho důvodu, aby načítání přehrávaných souborů z SD karty bylo bez jakýchkoli výpadků. Pokud jsou navíc odpovídající funkce opatřeny mutexy, je možné z karty střídavě číst a zapisovat dvěma nezávislými úlohami bez jakýchkoli dalších starostí.

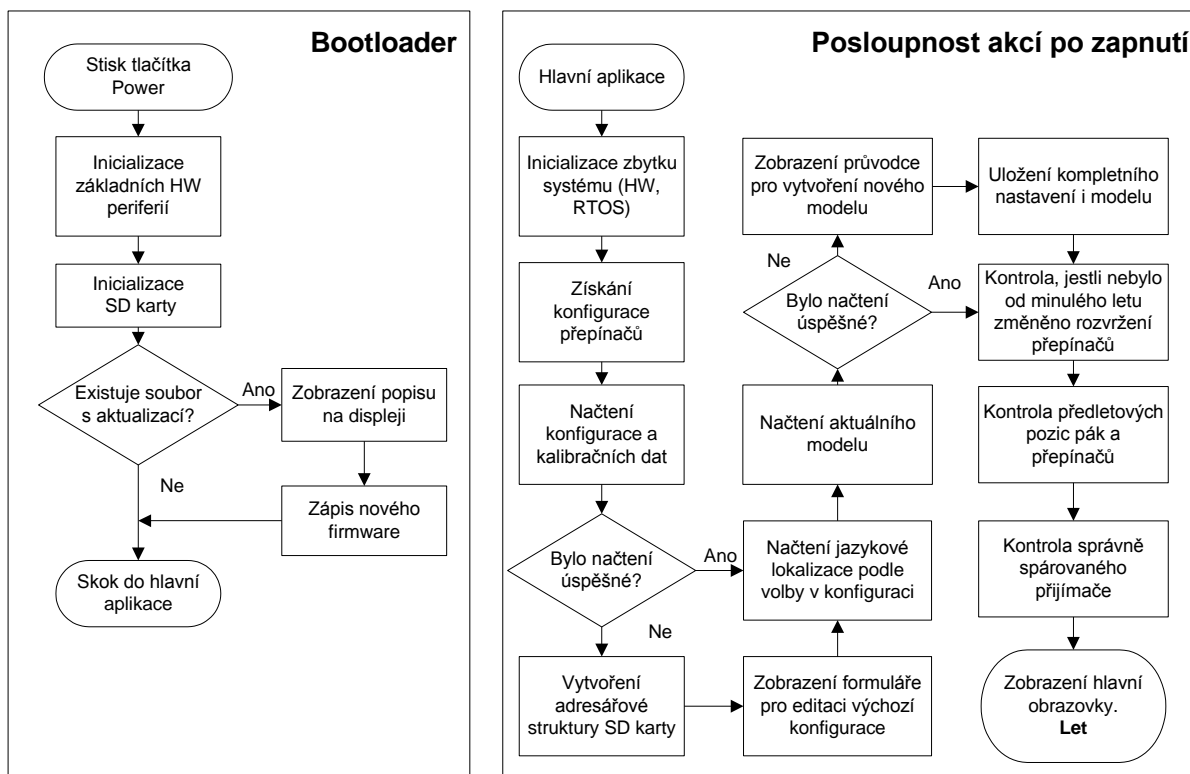
Třetí úloha (*ExecutiveTask*), mající nejvyšší prioritu, obsluhuje transformaci vstupů ovládacích prvků na výstupní pozice jednotlivých kanálů podle nastavení modelu. Pokud by došlo k jakémukoli výpadku ve zpracování této úlohy, pro ovládaný model by to mohlo skončit katastrofou. Z důvodu zvýšení bezpečnosti sem byla doplněna funkce watchdogu, který v případě selhání provede reset procesoru.

Poslední úlohou je *UsbTask*, jejímž úkolem je obsluha požadavků sběrnice USB tak, aby nemusely být prováděny během přerušení. Je to z toho důvodu, že tyto požadavky mohou být zpracovávány velice dlouhou dobu, a během přerušení nemá procesor možnost přepnout kontext, aby se mohl věnovat jiné, důležitější úloze.

6.2.1 Úkony po zapnutí

Na Obrázku 13 je znázorněn proces inicializace vysílače od zapnutí až do okamžiku, kdy je vše připraveno pro let. V první fázi se aktivuje integrovaný bootloader procesoru STM32F405, který je umístěn na prvních 64kB vnitřní Flash paměti. Ten provede inicializaci hardwarových komponent a na SD kartě vyhledá soubor **dc-16.bin** v adresáři **/Update**. V případě úspěšného nalezení je z tohoto souboru provedena aktualizace firmwaru hlavního programu. Jakmile bootloader svou činnost ukončí, je proveden skok na počáteční adresu sektoru, kde se nachází kód hlavní aplikace.

Následují standardní úkony načtení konfiguračních souborů do paměti RAM, validace vstupních údajů před letem a nakonec kontrola spárování se správným přijímačem. Ta je důležitá, neboť vysílací modul je v jednom okamžiku schopen komunikovat pouze s jedním přijímačem a jeho identifikátor se navíc musí nacházet v seznamu povolených přijímačů pro daný model. Je to víceméně bezpečnostní funkce, aby nemohlo dojít k tomu, že pilot ovládá jiný model, než má na vysílači aktuálně nastavený.



Obrázek 13: Posloupnost úkonů bootloaderu a hlavní aplikace v prvních okamžicích po zapnutí.

6.2.2 Transformační funkce

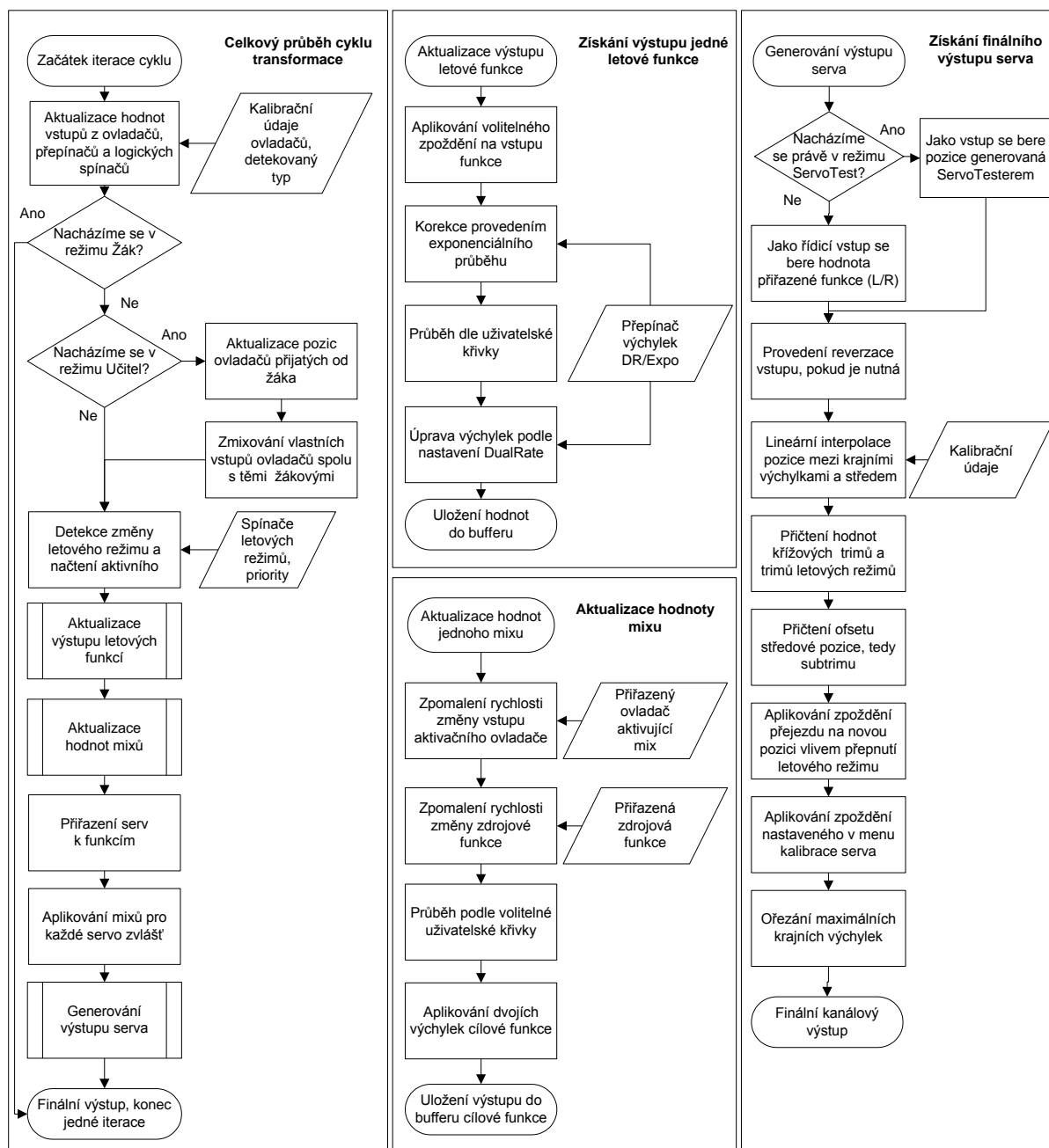
Transformační funkce je alfa a omega každého moderního modelářského vysílače. Jejím primárním úkolem je zpracovat veškeré vstupy pák, spínačů a proporcionálních ovladačů podle přesně daných pravidel a ve výsledku vrátit velikosti výchylek pro jednotlivá serva. Svou úlohu zde sice hrají veškeré komponenty zmíněné v kapitole 3, avšak obecně není tato funkce nikterak standardizovaná. Každý výrobce vysílačů má tuto funkci implementovanou odlišně a způsob implementace navíc nebývá zveřejněn. Z tohoto důvodu bylo při jejím návrhu nutné absolvovat rozsáhlé studium možností konkurenčních vysílačů (např. [10] a [11]), ale i tak ji bylo potřeba několikrát kompletně přepracovávat.

Implementace transformační funkce v podobě vývojového diagramu je zachycena na Obrázku 14. Ve stručnosti můžeme napsat, že máme n letových funkcí, každé z nich je dále přidělen jeden primární vstup a 0 až m ovládaných serv. Každé servo je možné zkalibrovat nezávisle na své zdrojové funkci. Při výpočtu finálních kanálových výstupů jsou pak vždy brány v potaz příslušné kalibrační údaje daného serva.

Nejprve je provedeno načtení aktuálních vstupních hodnot jednotlivých ovládacích prvků. Následně je zjištěno, v jaké konfiguraci pracují VF moduly (viz kapitola 5, kde je shrnuta operace v módu *Učitel/žák*). Jestliže máme zvolen jakýkoli jiný režim než *Žák*, procedura pokračuje ve vyhodnocení aktuálního letového režimu. Podle parametrů letových funkcí v něm uložených jsou dále aplikovány uživatelské křivky, exponenciální průběh a dvojí výchylky. Následně jsou vypočteny hodnoty mixů a ty jsou společně s výstupem odpovídající letové funkce předány příslušným servům.

Aplikace dvojích výchylek, křivek a mixů se vždy vztahuje k letové funkci, což znamená, že serva příslušející k jedné funkci budou mít vždy tyto parametry shodné.

V těch z následujících kapitol, jež se budou věnovat grafickému rozhraní programu a jednotlivým formulářům, jsou všechny zobrazené volby uzpůsobeny s ohledem na zde představenou transformační funkci a plně ji reflektují. Z tohoto důvodu je pro čtenáře nyní výhodné, aby si prošel další kapitoly, zejména ty o uživatelském rozhraní. K detailnějšímu studiu vývojového diagramu na Obrázku 14 je pak možné se kdykoli později vrátit.



Obrázek 14: Kompletní schéma transformační funkce s jejími podčástmi.

6.3 Navržené datové třídy

Použitý mikrokontrolér STM32 má k dispozici oproti PC velmi omezený výpočetní výkon a rovněž je třeba brát ohled na malou kapacitu RAM, která není ani v řádu megabajtů. Na jednu stranu jsou zde použity generické šablony jazyka C++ i dědičnost, na stranu druhou byla zablokována podpora výjimek v blocích *try* a *catch*. První volba velice urychlí a zpříjemní práci programátora a oprostí ho od nízkoúrovňových starostí; zablokování výjimek zase přinese velkou úsporu místa v přeloženém binárním kódu, přičemž je pouze nutné drobně upravit své návyky.

V aplikaci je využito dynamické alokace paměti pomocí standardních funkcí jako *malloc*, avšak v kontrastu k tomu nebylo možné použít standardní kontejnery jazyka C++, neboť jsou svou velikostí příliš objemné. Z tohoto důvodu bylo navrženo a implementováno v jazyce C++ několik základních tříd, které díky své zapouzdřenosti, autonomii a jednoduchosti použití výrazně urychlily vývoj za cenu pouze mírně zvýšených nároků na paměť.

6.3.1 Třída String

Třída String se používá pro správu a uchovávání textových řetězců. Obsahuje dynamicky alokované pole, které se vždy přizpůsobí tak, aby daný text pojalo. Práce s třídou je poměrně jednoduchá, pro účely projektu byly implementovány metody přiřazení řetězce, konkatenace s jiným řetězcem či formátovaným číslem. Je možné zarovnat text na daný počet znaků. Třída podporuje řetězce v kódování UTF-8, aby bylo možné dosáhnout vícejazyčnosti projektu.

6.3.2 Třída Vector

Existuje mnoho volně dostupných implementací obdobných tříd jako *std::vector*, ovšem jsou buď příliš paměťově náročné, nebo neobsahují přesně ty funkce, které jsou potřeba. Z tohoto důvodu byla implementována vlastní třída Vector, která je založena na práci s generickými datovými typy a správně funguje i s ukazateli. Instance této třídy vždy obsahuje dynamicky alokované pole prvků, ke kterému je možné přistupovat jako ke klasickému poli jazyka C. Jako doplněk můžeme prvky do Vectoru vkládat na libovolnou pozici. Pomocí přetěžování porovnávacích operátorů je možné prvky seřadit (použita byla metoda Shell Sort podle [17]) a následně v nich rychle vyhledávat díky použití algoritmu binárního prohledávání. Na architektuře Cortex-M4 zabírají instance třídy v paměti RAM o pouhých 15B navíc oproti prostému poli. To je akceptovatelná daň za plně automatickou správu paměti pro objekty uvnitř třídy.

6.3.3 Třída JSON

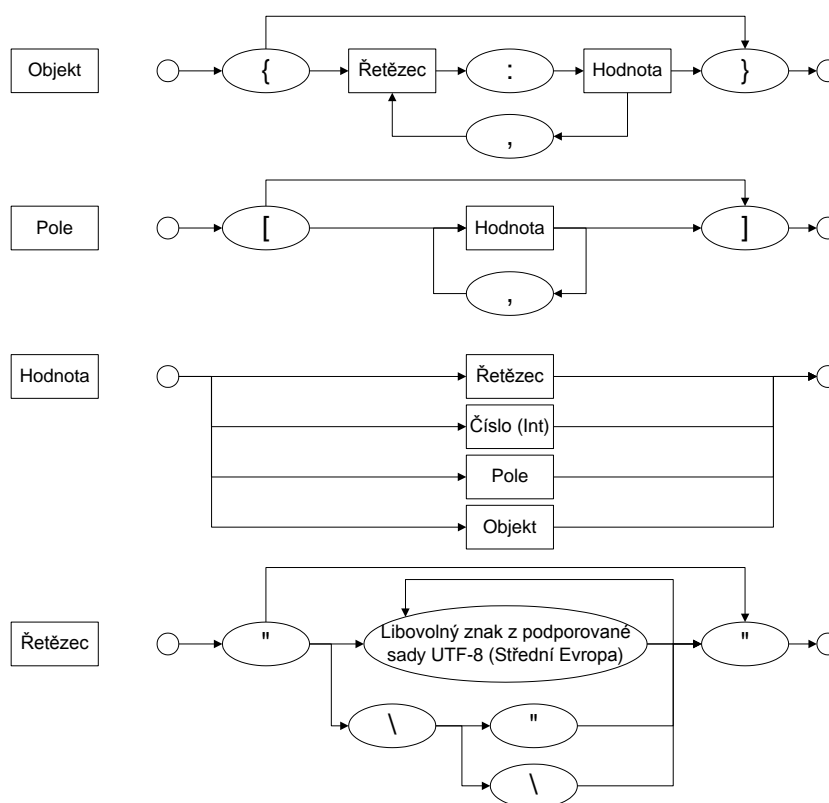
Veškeré konfigurační údaje vysílače jsou uloženy na SD kartě. Pro ukládání dat byl po předchozí analýze zvolen formát JSON (JavaScript Object Notation), který oproti YAML či XML více klade

důraz na jednoduchost zpracování před čitelností, avšak stále je snadné rozpoznat vnitřní strukturu z běžným pohledem. Je také standardním formátem používaným webovými aplikacemi.

Pro potřeby projektu byl učiněn malý průzkum mezi různými volně dostupnými implementacemi JSON parserů. Žádný však nevyhovoval, a to především z hlediska paměťové náročnosti, kdy například nebylo možné zpracovat soubor o velikosti 200kB při velikosti RAM pouhých 100kB. Proto bylo rozhodnuto o implementaci vlastního syntaktického analyzátoru.

Jeho gramatika je zobrazena na Obrázku 15 a je oproti standardní implementaci (viz [18]) zjednodušená, takže ze základních datových typů jsou přijímány pouze textové řetězce a celočíselné hodnoty.

Třída *Json* obsahuje hlavní veřejnou metodu *Parse*, jejímž úkolem je správně načíst a zpracovat daný dokument. Jako parametr je jí předán odkaz na funkci, která vždy provede naplnění vnitřního bufferu znaků, jestliže byl ten předchozí již zpracován. Jako další parametr metody *Parse* je možné předat odkaz s názvem souboru, jenž je funkci pro čtení znaků pokaždé předán. Posledním parametrem metody je callback funkce určené ke zpracování právě načteného prvku datového typu *Objekt*. Je možné procházet souborem velmi rychle a také po částech v malých dávkách. Rozhraní třídy umožňuje zahození dat částí načtené struktury, ale i kompletní stornování operace načítání dokumentu. Pokud zahodíme část datové struktury, bude jí obsazená paměť uvolněna a důsledku snadno projdeme celým dokumentem i s omezenou kapacitou paměti. Stornování načítání využijeme tehdy, když chceme z celého dokumentu zjistit pouze jednu konkrétní informaci – jakmile ji zjistíme, další zpracovávání textu je již zbytečné.



Obrázek 15: Použitá gramatika pro soubory typu JSON.

Třída `Json` dále dokáže dokumenty vytvářet a pomocí externího callbacku i okamžitě ukládat na disk. Při tvorbě dokumentu je prováděna zevrubná syntaktická kontrola pomocí zásobníku symbolů (implementován pomocí instance třídy `Vector`), vlastní obsah je rovnou ukládán po znacích na disk. V paměti se tedy nevytváří žádný datový strom. Jako ukázka použitelnosti rozhraní pro zápis může posloužit několik následujících řádků kódu (viz Tabulka 1).

Zdrojový kód	Výstup (<code>file.json</code>)
<pre> Json json; json.openFile(WriteCallback, "file.json"); json.openObject(); json.openObjectVar("Var1"); json.addString("Test"); json.closeObjectVar(); json.openObjectVar("Version"); json.addInt(12); json.closeObjectVar(); json.closeObject(); json.closeFile(); </pre>	<pre> { "Var1": "Test", "Version": 12, } </pre>

Tabulka 1: Ukázka použití třídy `Json` pro zápis do souboru.

6.3.4 Třída `Locale`

Protože vysílač bude primárně distribuován do zahraničí, byla lokalizovatelnost brána na zřetel již od prvních okamžiků vývoje. Veškeré grafické subsystémy hlavního programu proto podporují středo- a západoevropskou část znakové sady UTF-8. Je tedy možné psát texty opatřené diakritikou v klasickém textovém editoru a na displeji vysílače se vše korektně zobrazí.

Třída `Locale` má jednu globální instanci, která provádí inicializaci řetězců vždy po zapnutí vysílače, případně po změně jazyka v nabídce *Menu*→*Configuration*. Překlady jsou uloženy ve formátu JSON na SD kartě v souboru typu `/Lang/xx/lang.json`, kde `xx` značí kód jazyka. Jednotlivé položky jsou v tomto souboru umístěny vždy ve dvojici `<ID, "Přeložený text">`.

6.4 Přehrávání audia

Přehrávání audio souborů plní ve vysílači informační, bezpečnostní, ale také zábavní funkci. Později v této kapitole budou naznačeny způsoby použití ve všech třech oblastech, nyní však bude popsán způsob implementace akustického subsystému vysílače.

Základní třída pro práci se zvukovými soubory se jmenuje **`Audiocpp`**. Ta dokáže komunikovat s externím zvukovým kodekem po sběrnici I²C (konfigurace registrů) a I²S (samotná audio data). Umí dekodovat a přehrávat soubory s koncovkou WAV ve formátu mono i stereo, 8 bitů i 16 bitů na kanál. Umožňuje přehrávat až dva soubory současně, přičemž oba mohou mít jakoukoli vzorkovací frekvenci. Pokaždé totiž dochází k převzorkování na referenční frekvenci 44100Hz. K dosažení kvalitního výstupu při této operaci bylo použito metody *resample* z open source balíku Speex [19].

Třídě náleží také audio buffer o velikosti 32kB, do kterého jsou nahrávány aktuální části skladby. Buffer funguje na principu střídavého naplňování své horní a dolní poloviny, kdy v okamžiku přehrávání dat z jedné poloviny dochází automaticky k naplňování druhé poloviny čtením z SD karty. Je tak možno dosáhnout kontinuálního přehrávání souborů bez jakéhokoli snížení kvality. Velikost audio bufferu byla zvolena s ohledem na latence, které mohou vzniknout při současném čtení a zápisu na SD kartu – podle katalogových listů to může být až 200ms.

Třída `Audiocpp` využívá obsluhu přerušení při komunikaci s kodekem po rozhraní `I2S`. V této obsluze je mimo jiné prováděno i generování tónů, jež je aktivní např. vždy při stisku tlačítka.

6.5 Připojení k USB

Sběrnice USB (Universal Serial Bus) je moderní způsob, jak připojovat k počítači externí zařízení. Jelikož mikrokontrolér tuto sběrnici nativně podporuje a firma ST Microelectronics ke stažení na svých stránkách nabízí různé příklady jejího využití, byl jí vysílač vybaven. Zdrojové kódy výrobce procesorů byly přepracovány a zakomponovány do projektu.

Pokud ve vysílači aktivujeme funkci USB (viz nabídka *System*→*USB* v následující kapitole), zařízení se přihlásí jako takzvané *Composite Device*, kdy je jedna datová linka použita pro několik nezávislých rozhraní. V našem případě je implementována podpora protokolu *Mass Storage* pro pohodlné procházení adresářovou strukturou SD karty. Současně je zařízení dostupné jako osmiosý USB joystick vybavený 32 tlačítky, čehož je možné přímo využít pro ovládání počítačového leteckého simulátoru.

6.6 Grafické uživatelské rozhraní

V rámci diplomové práce bylo navrženo uživatelské rozhraní, které reflektuje požadavky uvedené v kapitole 4. Je určené pro monochromatický displej s rozlišením 320×240px a v jeho návrhu je zohledněno opakované používání základních grafických komponent, jako jsou výběrová pole nebo editory číselných hodnot.

Jako hlavní uživatelský vstup slouží rotační enkodér po pravé straně vysílače (viz Obrázek 12), pomocí něhož probíhá navigace strukturou menu ve vertikálním směru (hrubý výběr), v některých nabídkách dále i v horizontálním (jemný výběr). Pro potvrzování nebo stornování volby slouží tlačítka `Enter` resp. `Escape` nedaleko rotačního enkodéru. Dalších pět pomocných tlačítek pod displejem slouží jako rychlá volba pro vyvolání specifických akcí.


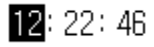

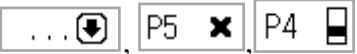





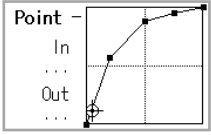


V následujících odstavcích bude popsáno několik klíčových obrazovek vysílače. Textové popisy na nich jsou uvedeny v anglickém jazyku, aby byly v konzistenci s aplikací přiloženou na doprovodném CD.

6.6.1 Základní formulářové komponenty

Při návrhu komponent formulářové struktury bylo často využíváno dynamických vlastností jazyka C++ jako jsou dědičnost a polymorfismus. Za cenu mírně zvýšených nároků na paměť bylo možné vytvořit grafický subsystém, který za použití základních stavebních kamenů zvládne vytvořit téměř libovolnou strukturu uživatelských nabídek.

Základní komponentou grafického subsystému je především globální třída **Display**, která v sobě zapouzdřuje veškeré nízkoúrovňové operace s displejem. Provádí např. vykreslování grafických primitiv do absolutních souřadnic displeje nebo do vyznačeného omezeného prostoru za použití relativních souřadnic. Tato třída si uchovává stav obrazovky ve vnitřním bufferu (cca 10kB), takže provedení operací je vždy velmi rychlé. Jakmile jsou všechny grafické úkony dokončeny a je tak aktualizována celá obrazovka, následuje přeposlání kompletního bufferu do fyzického displeje, kde dojde k vizualizaci údajů.

Na třídu Display plynule navazuje třída **FormManager**. Ta je vytvořena jako singleton, takže je možné vytvořit maximálně jednu instanci a na ni se lze odkudkoli z aplikace odvolávat. Tato třída je vlastně jakýsi zjednodušený okenní manažer, stará se o obsluhu stisku tlačítek a předávání událostí mezi komponentami, dále volá přednastavené periodické události formulářů v zadaném intervalu.

Název třídy	Ilustrace	Název třídy	Ilustrace
Row		Timebox	
Link		Inputbox	
Label		Textbox	
Selectbox		Buttonbox	
Intbox		Curvebox	
Datebox		CurveSmallBox	

Tabulka 2: Přehled formulářových komponent použitých v aplikaci.

Interně jsou v instanci této třídy uchovávány všechny spuštěné formuláře jako na zásobníku. Je tak možné rekurzivně v cyklu vytvářet další a další formuláře až do okamžiku, kdy dojde paměť a program havaruje. Každý spuštěný formulář musí být potomkem třídy **Form**. Tímto omezením bylo dosaženo jednotného rozhraní všech formulářů v celé aplikaci (je jich zde celkově téměř sto). Třída **Form** obsahuje virtuální metody pro obsluhu události zmáčknutí tlačítka, vykreslení formuláře nebo pravidelné spouštění v určitém intervalu. Rovněž je do ní možné přidat sadu interaktivních komponent, které jsou zděděné ze třídy **FormComponent** a slouží pro rychlou, jednoduchou a bezpečnou navigaci v nabídkách, popř. editaci různých hodnot. Některé komponenty jsou vyjmenovány v Tabulce 2.

Hlavní komponentou formuláře je vždy (maximálně) jedna instance třídy **Frame** (není zobrazeno v Tabulce 2). Do této instance se pak přidávají další komponenty, které jsou dále zobrazovány ve zvoleném rozvržení. Třídě **Frame** je nyní možné nastavit jednosloupcový nebo dvousloupcový layout, přičemž je umožněno vertikální rolování, pokud velikost formuláře přesáhne velikost displeje.

Třída **Row** slouží k navigaci mezi horizontálně rozmístěnými komponentami. Lze jí přiřadit až osm podkomponent, mezi nimiž uživatel vybírá po aktivaci daného řádku.

Třídy **Frame** a **Row** můžeme označit jako komplexní komponenty. Zbývající třídy jsou relativně jednodušší, neboť ve své struktuře již nemohou obsahovat žádné další zanořené komponenty. Slouží především pro provedení nějaké akce nebo editaci dané položky. K usnadnění navigace uživatele slouží třída **Label**, která pouze zobrazuje zadaný text jako konstantní popisek, nebo třída **Link**, jež po najetí a stisku rotačního enkodéru vytvoří předem zadaný formulář. Je tak relativně snadné procházet celou strukturou nabídek.

Další grafické komponenty již slouží k zobrazování, ale také editaci hodnot jim přidělených položek. Datové typy editovaných hodnot je možné odvodit z názvu komponenty. **Intbox** tedy slouží k editaci čísel s pevnou desetinnou čárkou (je možné zvolit z několika grafických vzhledů), **Selectbox** je možné vybírat z několika předpřipravených voleb. Komponenta **Textbox** dokáže editovat textové řetězce, **Datebox** a **Timebox** zase datum a čas. **Buttonbox** slouží k zobrazení interaktivní bitmapové grafiky a po kliknutí na něj lze provést libovolnou akci.

Mezi specifické komponenty patří poslední tři, tzn. **Inputbox**, **Curvebox** a **CurveSmallBox**. První jmenovanou uživatel vybírá a přiřazuje libovolný fyzický ovládací prvek. Když je potřeba například přiřadit spínač k řízení vysouvání a zatahování podvozku, je to možné pomocí této komponenty. **CurveSmallBox** a **Curvebox** slouží k zobrazení, resp. editaci lineárně aproximované křivky. Komponenta na vstupu dostane seznam kontrolních bodů, jimiž lze volně pohybovat, a které dohromady vytváří spojitou po částech lineární křivku. Je možné volit z několika předvoleb nebo si vytvořit křivku vlastní.

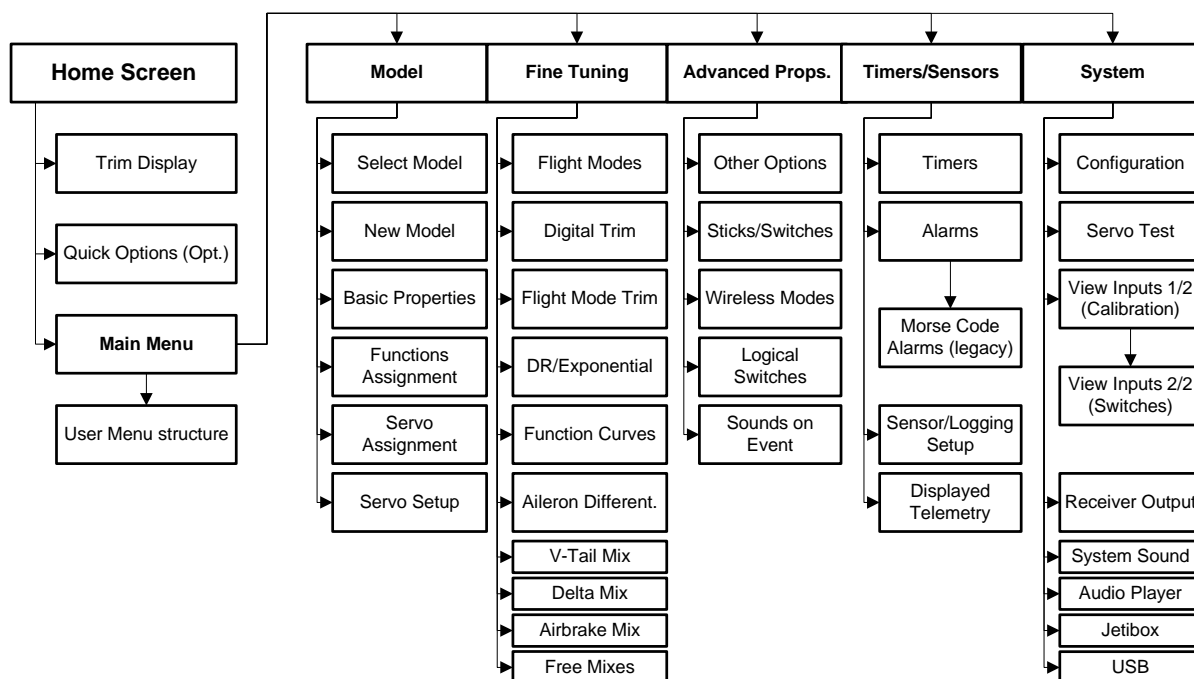
6.6.2 Struktura menu

Na Obrázku 16 je zobrazena kompletní struktura nabídek vysílače. Menu bylo uspořádáno do logických celků tak, aby významově podobné obrazovky byly uvedeny vždy ve stejné nabídce, čímž vzniklo několik kategorií, které jsou přístupné přes Hlavní menu.

V kategorii *Model* jsou vedle sebe umístěny položky, které jsou spojené s konkrétním modelem a jejich nastavování se provádí ideálně pouze jednou při zakládání nového modelu. Jestliže nastavujeme model podle návodu, stačí vyplnit formuláře v této kategorii a bez dalšího zdržování s ostatními nabídkami je možné jít létat. Údaje jsou platné pro celý model a všechny jeho letové režimy, dále mají vliv na strukturu některých jiných nabídek.

Kategorie *Fine Tuning* (jemné ladění) poskytuje správu letových režimů a veškeré údaje se zde nastavují pro každý režim nezávisle. Vlastnosti modelu jsou upravovány tak, aby jeho řízení bylo pohodlné a příjemné. To je však u každého pilota a modelu silně individuální, takže počet podnabídek a modifikovatelných proměnných je zde poměrně mnoho. Obsah této kategorie se však vždy přizpůsobuje konfiguraci aktuálního modelu, takže například u modelu vrtulníku nelze nastavit aerodynamickou brzdu, jež patří výhradně pod doménu plošníků.

V menu *Advanced Properties* je možné volit z dodatečných nabídek, jejichž nastavování má vliv na celý model, ovšem volby nejsou tak kritické, aby bez nich nešlo létat či jinak řídit.



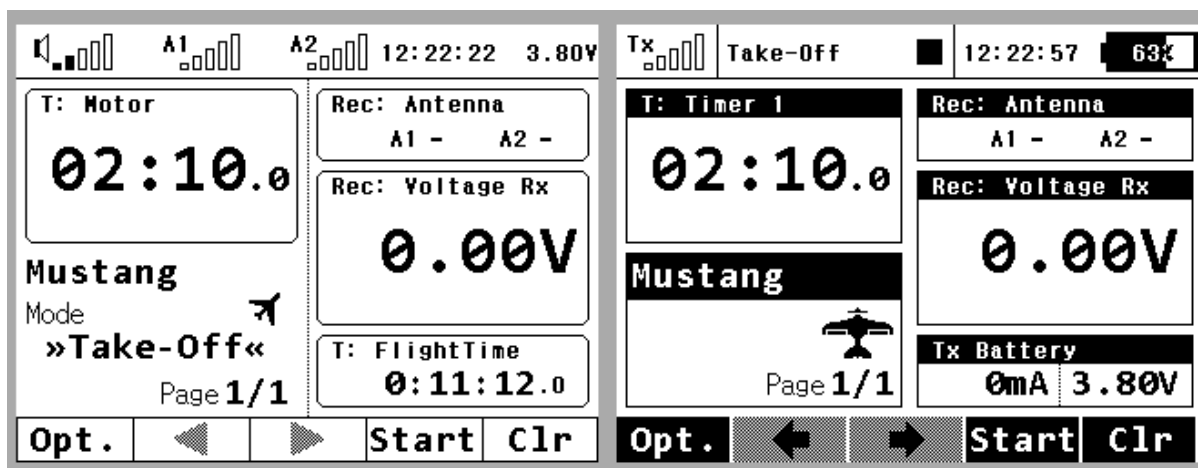
Obrázek 16: Kompletní struktura nabídek uživatelského rozhraní.

Kategorie *Timers/Sensors* shrnuje doplňkové funkce týkající se časovačů, alarmů a telemetrie obecně. Volí se zde také, které proměnné se budou ukládat na SD kartu pro pozdější vyhodnocení, tzv. „bezdrátová černá skříňka“.

Poslední kategorie je nazvána *System*, neboť shrnuje parametry a funkce platné pro celý vysílač bez ohledu na zvolený model. Je zde možné provést základní konfiguraci systému nebo kalibraci ovladačů. Je zde též možné aktivovat některé doplňkové funkce zařízení, jakými jsou *Audio Player* nebo připojení k *USB*.

Vybrané nabídky budou podrobně popsány v následujících odstavcích.

6.6.3 Hlavní obrazovka



Obrázek 17: Dvě varianty grafické úpravy hlavní obrazovky. Pro finální implementaci byla použita ta napravo.

Na Obrázku 17 jsou vykresleny dva grafické návrhy hlavní obrazovky. Ta by měla být přehledná a informativní, aby se v ní pilot rychle orientoval a dokázal zjistit potřebné údaje – obrazovka jej totiž provází během velké části letu. Oba dva styly byly postaveny proti sobě při výběru toho vhodnějšího, a jak je vidět z dalších snímků, nakonec byl použit více „kontrastní“ styl zobrazený vpravo.

V jeho horním pruhu se nachází indikátor intenzity příjmu signálu, dále aktuální letový režim, zobrazení času a procentuální vybití baterie vysílače. Obrázek vlevo má v horním pruhu navíc indikátor hlasitosti reproduktoru a ukazatele příjmu signálu jsou zdvojené, aby byl zachycen pohled na obě antény. Ukázalo se však, že stačí jednoduché zobrazení signálu, jež podává celkový přehled. Dotázání uživatelé navíc upřednostňovali textový popis letového režimu před informací o hlasitosti.

Středová oblast hlavní obrazovky slouží jako velké konfigurovatelné pole, kde je možné v libovolném pořadí zobrazit jakýkoli dostupný panel telemetrie, časovačů apod. Takto je možné vytvořit podle potřeby i několik kompletně zaplněných obrazovek a během letu mezi nimi přepínat. Nastavení údajů na hlavní obrazovce se pak provádí pomocí menu *Timers/Sensors→Displayed Telemetry*. Na hlavní ploše vlevo dole jsou konstantně zobrazeny další informace týkající se modelu – jde především o název modelu a doplňkové stavové popisy, které nyní nejsou viditelné.

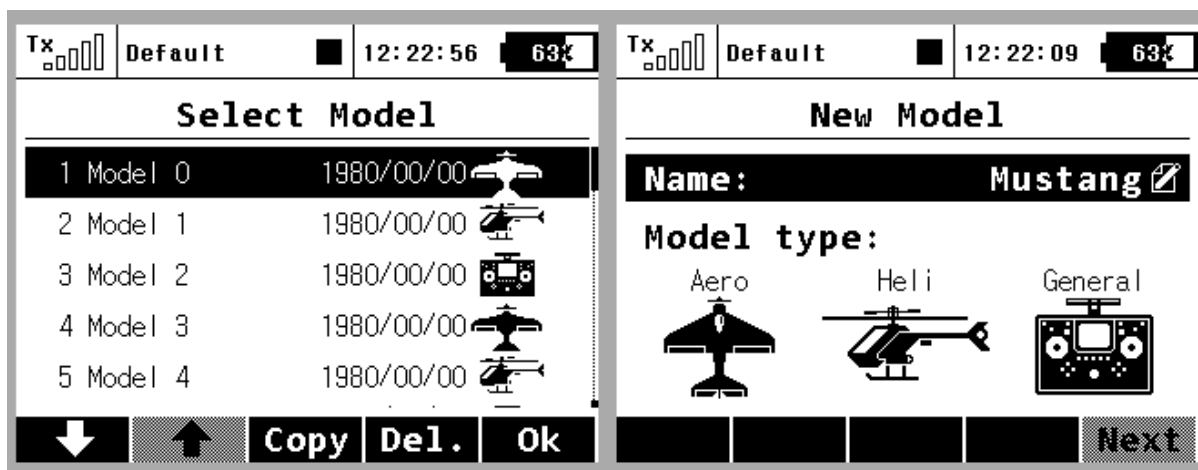
Spodní pruh je určen pro grafické popisky tlačítek, jež mohou být textové i bitmapové. Na hlavní obrazovce slouží tlačítka postupně k vyvolání rychlého menu (nastavení hlasitosti apod.), pro přepínání mezi obrazovkami telemetrických polí a pro řízení časovačů.

6.6.4 Základní nastavení modelu

Do základního menu se z hlavní obrazovky dostaneme stisknutím tlačítka *Menu* na vysílači. Mezi kategoriemi je možné pohodlně procházet pomocí rotačního enkodéru, jehož stisk znamená potvrzení volby, tedy i otevření nového formuláře. Do předchozí úrovně výpisu kategorií se vracíme tlačítkem *ESC*.

V nabídce *Model*→*Select Model* (Obrázek 18 vlevo) je možné listovat seznamem již vytvořených modelů. Tento seznam je vytvářen dynamicky podle obsahu souborů uložených na SD kartě v adresáři **/Models**. Název souborů je tvořen z pořadového čísla modelu (počáteční čtyři znaky) a dále z prvních písmen názvu modelu. Bohužel nebylo možné použít delší název než klasický 8+3 z důvodu patentových praktik Microsoftu, takže zběžný pohled při otevření adresáře **/Models** normálnímu uživateli o uloženém modelu mnoho informací neposkytne.

Načtení seznamu modelů probíhá v několika etapách, především z důvodu nedostatku paměťového prostoru pro obsáhnutí kompletního adresáře. V první etapě je proveden průzkum složky **/Models**, kdy je zjištěno nejvyšší dostupné pořadové číslo modelu. Dále je prováděn výběr a řazení dostupných modelů do seznamu o x prvcích (cca 20), od počátečního indexu y . Řazení probíhá pomocí metody Insert Sort na základě pořadového čísla v názvu souboru. Tímto způsobem se projde celým adresářem, navíc dochází k automatickým opravám názvů souborů při detekci nesrovnalostí. Výsledných x souborů je postupně načteno a zpracováno třídou Json, takže jsou nakonec zjištěny i údaje jako kompletní název modelu nebo jeho typ. Zobrazení do grafické podoby je pak již jednoduché s využitím obecných formulářových komponent.

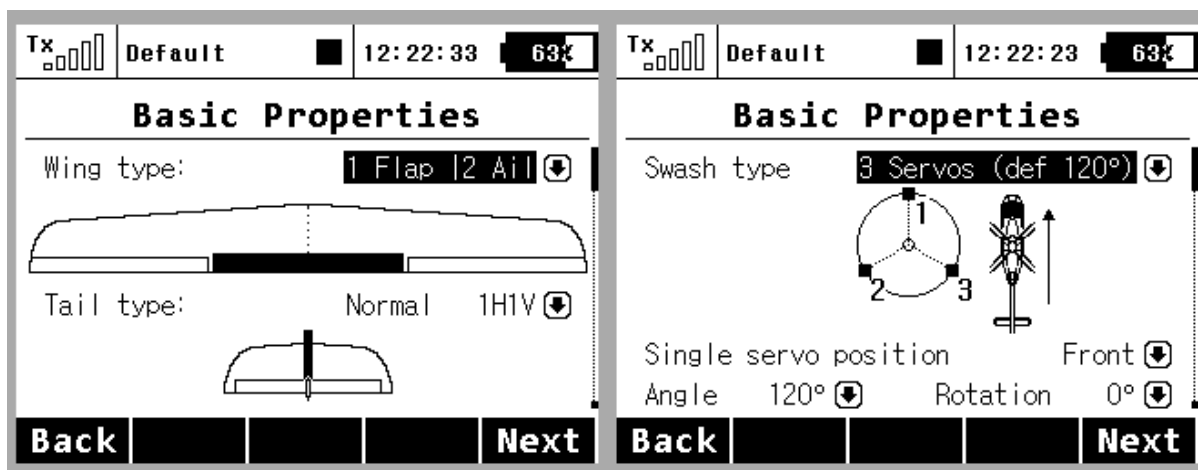


Obrázek 18: Zleva obrazovka pro výběr modelu a založení nového.

Na Obrázku 18 vpravo se nachází obrazovka pro vytvoření nového modelu. Zde je možné zvolit jeho název, podle kterého jej bude možné identifikovat a také se od něj bude odvíjet výsledný název konfiguračního souboru na SD kartě. Dále je třeba zvolit typ modelu. Na výběr jsou položky *Aero* a *Heli*, které v sobě zahrnují mnoho specifických předvoleb (viz kapitola 3), a potom je tu obecný typ *General*, jenž nemá žádné vlastnosti přednastavené a vše je nutné zadat manuálně. Tento typ využijí především stavitelé řízených modelů lodí nebo stavební techniky, kdy je potřeba mnoho ovládaných funkcí, avšak letecké předvolby jsou spíše na škodu.

Stisknutím tlačítka *Next* dojde ke spuštění průvodce, po jehož dokončení budeme mít nakonfigurovaný nový model. Nyní je zobrazena jedna z variant na Obrázku 19 v závislosti na zvoleném typu modelu. Jestliže máme nastaven typ *Aero*, je potřeba zadat počet serv v křídle (rozlišeno na vztlakové klapky a křídélka), typ ocasních ploch, počet motorů a serv ovládajících podvozek. Podle těchto parametrů budou dále automaticky vytvořeny příslušné letové funkce a k nim budou přiřazeny i ovladače.

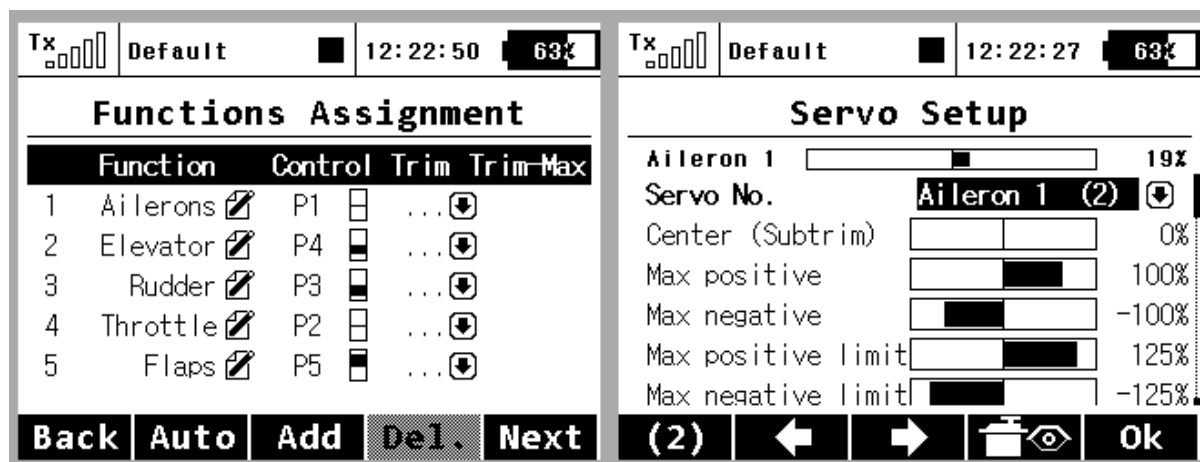
Pokud máme zvolen typ *Heli*, aplikaci musí být sděleny detailní informace o způsobu mechanického provedení rotorové hlavy (viz kapitola 3). V případě špatného zadání by totiž vrtulník byl kompletně neovladatelný, mohlo by dojít i k jeho trvalému poškození. Mezi význačné parametry, které se zde zadávají, patří zejména typ cyklyky, jež může být tříservová nebo čtyřservová. U tříservové se dále nastavuje úhel svíraný mezi servy č. 1 a č. 2, rotace celého soustrojí kolem vertikální osy a příznak čelní pozice serva č. 1, tj. zda se nachází před ostatními servy ve směru letu nebo až za nimi.



Obrázek 19: Základní informace o konstrukci modelu. Vlevo obrazovka pro plošníky, vpravo pro vrtulníky.

Po stisknutí tlačítka *Next* přejdeme ve všech do nabídky *Functions Assignment* (Obrázek 20), tedy přiřazení funkcí. Zde je vhodné především zkontrolovat, jestli letovým funkcím odpovídají správné ovládací prvky. U typů *Aero* a *Heli* je toto přiřazení automatické, u typu *General* si uživatel sám musí všechny funkce vytvořit – k tomu obecně slouží tlačítko *Add*.

Je možné vytvořit až 16 libovolně pojmenovaných funkcí a k nim dále přiřadit libovolný ovladač. Jedna letová funkce může obsluhovat volitelný počet serv, obecně 0 až 16 podle počtu kanálů. Výchozí konfigurace funkcí se řídí podle parametrů nastavených v menu *Basic Properties*.



Obrázek 20: Vlevo přiřazení vstupů k ovládaným funkcím, vpravo kalibrace křídélkového serva.

Po stisknutí tlačítka *Next* a následném potvrzení aktivace modelu průvodce končí na poslední obrazovce nazvané *Servo Setup* (Obrázek 20 vpravo). Zde již má uživatel zapnutý model a po jednom nastavuje parametry serv. U nich lze měnit směr pohybu (reverzace), velikost výchylky (max. positive/negative), středovou pozici (subtrim), ale i dobu přeběhu z jednoho krajního bodu do druhého, více viz kapitola 2.

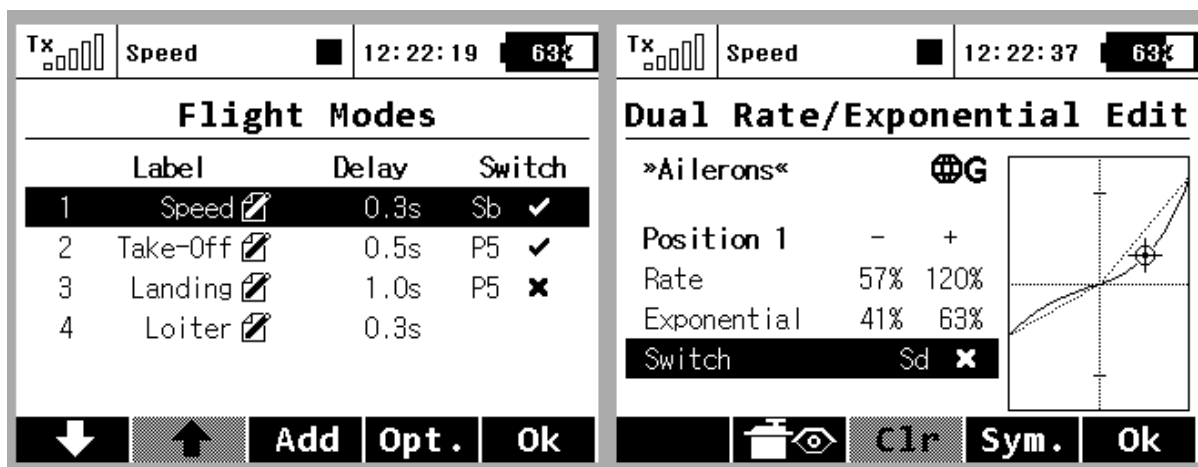
6.6.5 Nastavení pro letové režimy

Po vytvoření jednoduchého modelu je většinou možné provést první let. Serva se totiž hýbou správně a přiřazení ovladačů odpovídá zvyklostem modeláře. Složitější modely však vyžadují další preciznější úroveň nastavování. Je potřeba např. seřadit brzdicí klapky, správnout pohyb křidélek dohromady se směrovým kormidlem a podobně.

U maket a také závodních modelů je kladen důraz na používání letových režimů. Na Obrázku 21 vlevo je znázorněno, jak může seznam těchto režimů typicky vypadat (menu *Fine Tuning*→*FlightModes*). Každý režim má unikátní jméno, jež se zobrazuje v horním informačním pruhu pro ten mód, který je právě aktivní. Ve vysílači je možné vytvořit až deset nezávislých režimů. Jejich parametry lze editovat pomocí nabídek v kategorii *Fine Tuning*.

Ve formuláři *Flight Modes* můžeme letovým režimům přiřadit zpoždění, pokud nechceme, aby vzájemné přepínání bylo cukané. Řádky v tomto menu jsou seřazeny podle svých priorit, takže ten, jenž má prioritu nejvyšší, je zobrazen rovněž nejvýše. Úplně spodní letový režim je definován jako výchozí a je zde proto, aby nemohla nastat situace, kdy by nebyl žádný režim vybrán. Proto mu není ani možné přiřadit aktivační spínač. Ostatním je možné tento spínač libovolně přiřadit a platný režim

je pak právě ten, který má svůj spínač ve stavu *Zapnuto* a jeho priorita je nejvyšší. Na Obrázku 21 je to názorně ukázáno – aktuálně aktivním je režim Speed.

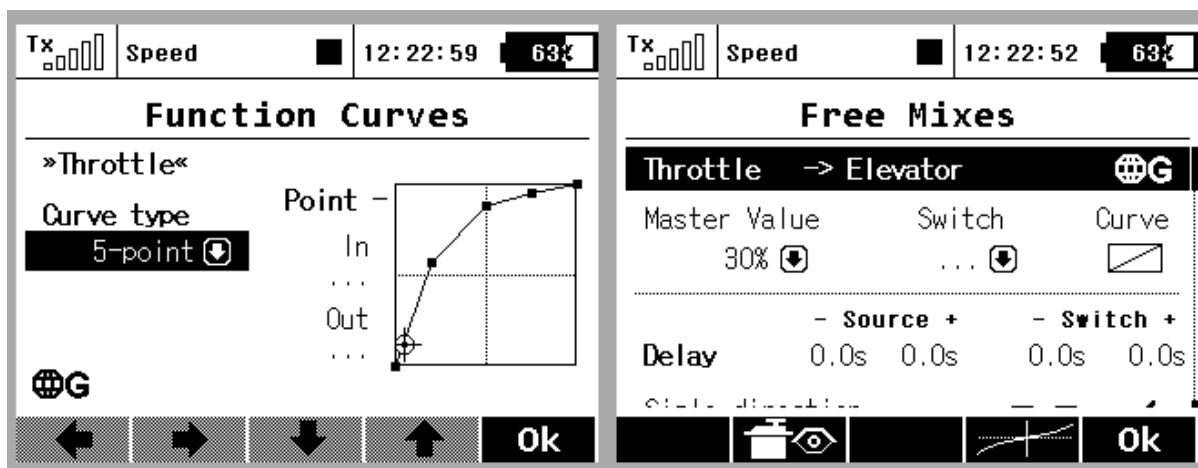


Obrázek 21: Vlevo seznam vytvořených letových režimů, vpravo editace DR/Expo u křídélek.

Neméně důležitou položkou oproti letovým režimům je možnost nastavení exponenciální odezvy a dvojích výchylek – viz Obrázek 21 vpravo. To bylo podrobně popsáno v kapitole 3, zde je možné doplnit, že vysílač umožňuje nezávislé aplikování exponenciálního průběhu na horní i dolní stranu výchylky serva. V této nabídce je také možné přiřadit přepínač, který mezi jednotlivými výchylkami vybírá. Je tak docíleno, že reakce modelu přesně a velmi jemně odpovídají tomu, co pilot žádá, dokonce i s ohledem na povětrnostní podmínky.

V kategorii *Fine Tuning* se nachází ještě velké množství podnabídek, jejichž popis by si dále vyžádal mnoho prostoru, avšak není úkolem diplomové práce, aby sloužila jako uživatelský manuál. Z tohoto důvodu jsou zde popsány pouze následující dvě obrazovky, neboť jejich pochopení je zásadní pro využívání kompletních možností vysílače.

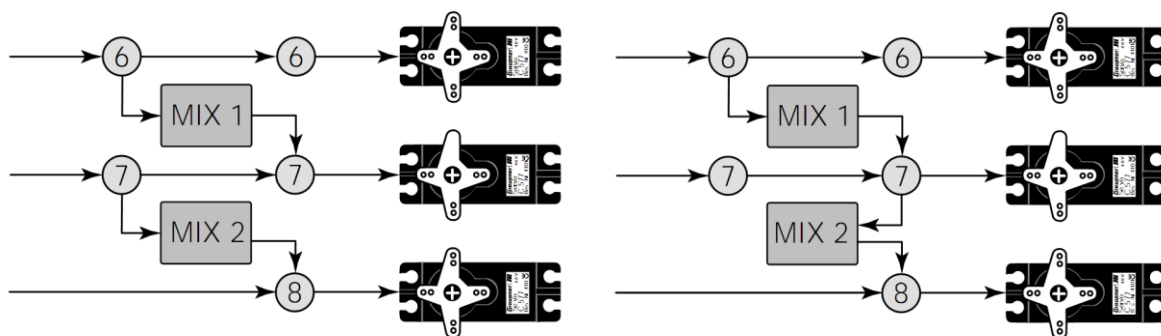
První nabídka, nazvaná *Function Curves* (na obrázku 22 vlevo), slouží obdobně jako menu *Dual Rate/Exponential* k vložení nelineární odezvy při reakci na změnu polohy ovladače. Tady je však aplikován jiný přístup. Je totiž možné zvolit až devět kontrolních bodů a pomocí nich vytvořit libovolnou křivku požadovaného tvaru. Tato křivka je pak při pohybu ovladače mezi kontrolními body lineárně interpolována. Pomocí zvýrazněného *Selectboxu* je navíc možné vybrat z několika předvoleb, jakými jsou například absolutní hodnota vstupu nebo kladný či záporný skok při přechodu nulou.



Obrázek 22: Vlevo křivka rozběhu otáček motoru, vpravo obrazovka s volným mixem potlačení výškovky při přidání plynu.

Poslední představenou nabídkou z kategorie *Fine Tuning* je obrazovka *Free Mixes*, jež je ilustrována na Obrázku 22 vpravo. Pokud chceme dosáhnout netriviálního chování modelu v reakci na řídicí vstupy, je pochopení možností mixování jednotlivých funkcí mezi sebou takřka nutností. V této nabídce se volí zdrojová a cílová funkce, přičemž zdrojová je brána jako vstup mixu. Po aplikaci mixu se do cílové bude funkce proporcionálně promítat vliv řídicí funkce, dojde tedy k jejímu jakémusi spřažení. Podstata mixů je znázorněna na Obrázku 23. V podstatě existují dvě varianty mixování a uživateli by mělo být umožněno si mezi nimi volně vybírat. První variantu představuje lineární mixování, kdy jsou jako vstupy brány pouze aktuální hodnoty funkcí a výstupy mixů pokračují rovnou do serv.

Pokud například budeme nastavovat jeden mix, který při zvýšení otáček motoru provede potlačení výškového kormidla (z důvodu zvýšeného klopivého momentu) a k němu vytvoříme ještě další mix, jenž při pohybu výškovkou provede vysunutí vztlačových klapek, pak při lineárním mixování nikdy nenastane situace, že by efekt přidání otáček motoru měl jakýkoli vliv na pohyb vztlačových klapek. Zde totiž nelze uvažovat žádné tranzitivní propojení mixů mezi sebou.



Obrázek 23: Vizualně znázorněná podstata aplikování mixů. Vlevo lineární mix, vpravo mix s možností zacykleného vstupu. Převzato z [10].

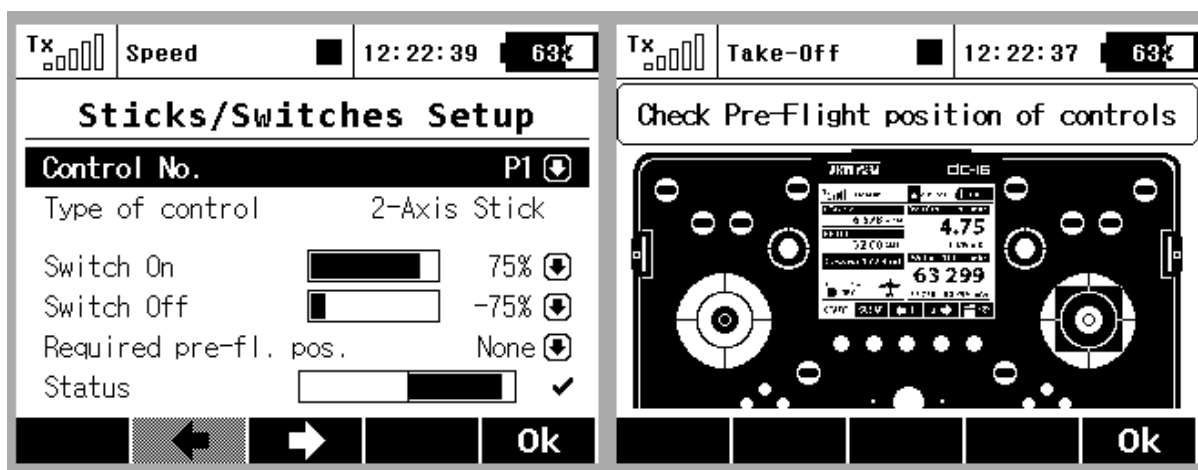
Druhý typ mixů je o něco složitější, protože vyžaduje vnitřní stavový buffer s hodnotami vstupů a výstupů použitých v předchozí iteraci výpočtu (Obrázek 23 vpravo). Avšak na druhou stranu je jím umožněn tranzitivní přenos mixované hodnoty mezi jednotlivými kanály. V našem příkladu se tedy při zvýšení otáček motoru současně potlačí výškovka a také se pohnou vztlakové klapky.

Ve vysílači jsou ve výchozím stavu všechny vytvořené mixy označené jako lineární. Aby mohly sloužit v pokročilejším režimu, je nutné povolit položku *Link* na formuláři *Free Mixes*.

6.6.6 Další volby vztažené k modelu

V kategorii *Advanced Properties* se nacházejí dodatečné volby modelu, které se převážně týkají pokročilejších nastavení ovládacích prvků. Na Obrázku 24 vlevo je pro ilustraci uvedena nabídka *Sticks/Switches setup*, kde je možné nechat si zobrazit postupně všechny ovladače a přepínače. U proporcionálních ovladačů je možné nastavit dvě úrovně prahování tak, aby je bylo možné používat i jako spínače těch funkcí, které nedovolují mít proporcionální vstup. Výstupem prahování je pak jedna ze tří hodnot $\{-1, 0, 1\}$ podle pozice ovladače.

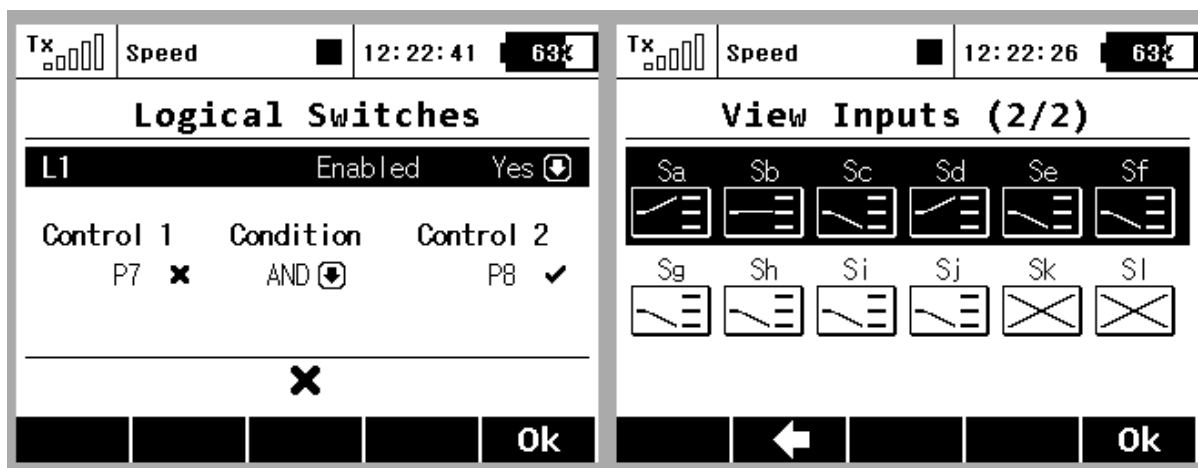
Nabízené možnosti v menu *Sticks/Switches Setup* jsou různé pro odlišné typy vstupních prvků. U typu *Tlačítko* lze např. nastavit mód s pamětí, tedy aby se chovalo jako bistabilní klopný obvod. V tomto menu je rovněž možné zvolit vyžadovanou předletovou pozici spínačů nebo pák. Pokud posléze model načteme, bude zobrazován informační formulář z Obrázku 24 vpravo, jenž sdělí, které prvky porušují předletovou konfiguraci. Formulář je pak automaticky skryt, jestliže předletová kontrola ohlásí splnění všech podmínek. Do té doby však je řízení modelu kompletně odpojené, aby se předešlo některým nechtěným stavům, například samovolnému roztočení motoru.



Obrázek 24: Nastavení prahů při transformaci proporcionálního ovladače na třípolohový přepínač, vpravo je zobrazena ilustrace, která se zobrazí po aktivaci volby *Required preflight position* a novém načtení modelu.

Obrazovka *Logical Switches* je také součástí kategorie *Advanced Properties* (Obrázek 25 vlevo). Tady se výrazně rozšiřují možnosti ovládacích prvků, neboť je možné vytvořit tzv. virtuální pře-

pínač, který bere na svém vstupu hodnoty dvou libovolných ovladačů a následně je vyhodnotí podle zvolené logické podmínky. Je možné zvolit režim *And*, *Or* nebo *Multi*. Podle názvů prvních dvou je jasný i jejich význam, třetí operátor slouží ke zkombinování dvou dvoupolohových přepínačů a má za výsledek vytvoření jednoho třípolohového. V Tabulce 3 je znázorněna výsledná logická hodnota. Ta zde není binární, ale je symetrická vzhledem k nule, neboť výstupy přepínačů jsou rovněž chápány jako maximální možné výchylky proporcionálních ovladačů.



Obrázek 25: Vlevo obrazovka logických, tzn. virtuálních spínačů. Vpravo vizualizace poloh fyzických přepínačů.

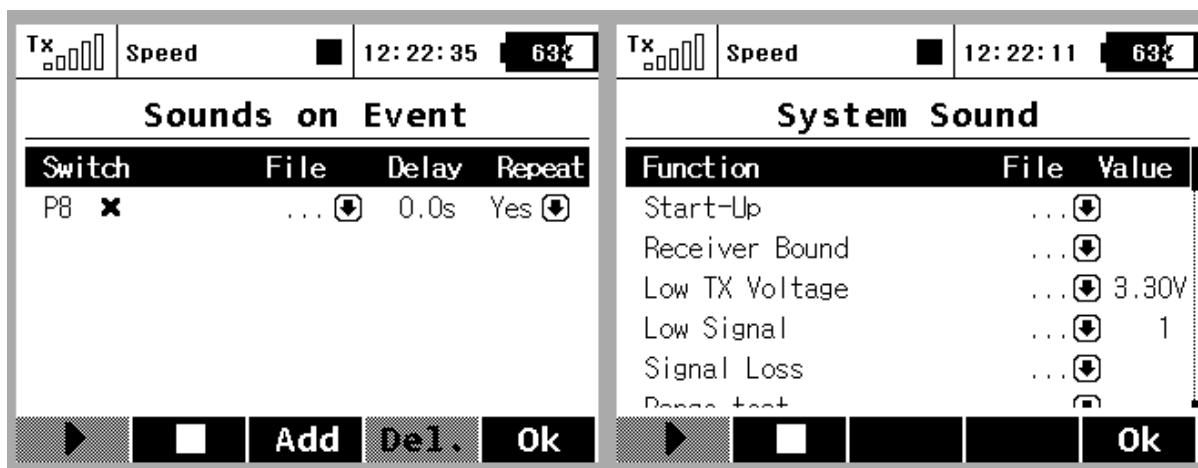
Spínač 1	Spínač 2	Operátor And	Operátor Or	Operátor Multi
-1	-1	-1	-1	-1
-1	1	-1	1	0
1	-1	-1	1	0
1	1	1	1	1

Tabulka 3: Pravdivostní tabulka aplikovaných operátorů logického spínače.

6.6.7 Možnosti přehrávání audio souborů, alarmy a časovače

Vysílač Jeti DC-16 nabízí mnoho možností přiřazení zvukové signalizace k určitým akcím, které jsou buď podmíněné pozicí určitého spínače, nebo hodnotami přijatých telemetrických dat. Je tak docíleno stavu, že pilot vždy ví, v jaké kondici se model nachází a může se plně soustředit na pilotáž. Současně jej nic nenutí přeastřovat svůj zrak displej a riskovat tak ztrátu modelu.

V nabídce *System*→*System Sound* (Obrázek 26 vpravo) je možné přiřadit systémové zvuky, které jsou platné pro celý vysílač. Jde především o různé druhy varování, přičemž nejdůležitější jsou varování o nízkém napětí baterie vysílače a informace o slabém příjmu signálu. Každé události můžeme přiřadit libovolný soubor typu WAV z adresáře **/Audio** na SD kartě.

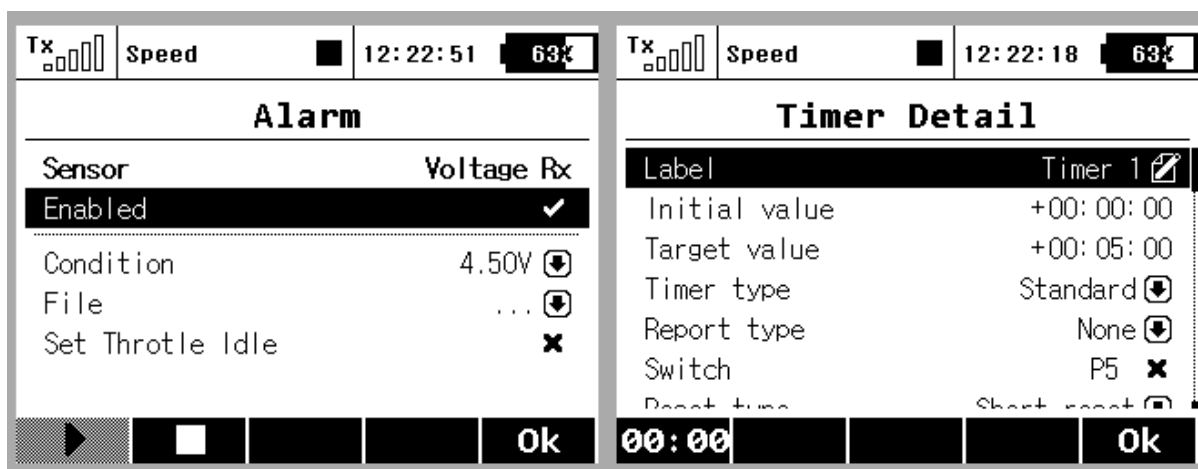


Obrázek 26: Vlevo přiřazení spuštění audio souboru nějaké pozici přepínače, vpravo nabídka systémových zvuků.

V nabídce *Advanced Properties*→*Sounds on Event* je možné přiřadit vlastní zvuky k libovolné akci na základě sepnutí spínače. Tyto zvuky jsou pak spíše informativního charakteru, neboť neposkytují žádnou zpětnou odezvu z modelu. Na druhou stranu je lze s výhodou použít např. při přepínání letových režimů tak, aby pilot bez pohledu na displej věděl, který režim je právě aktivní. Při vysouvání klapky či podvozku je zvuková signalizace také použitelná, neboť můžeme předejít situacím, kdy je některá funkce spuštěna omylem.

V nabídce *Sounds on Event* se k přehrávanému souboru přiřazuje spouštěcí přepínač a dále je možné zvolit časové zpoždění, s jakým bude soubor přehrán po aktivaci tohoto přepínače. Nakonec zaškrtneme položku *Opakování* v případě, že potřebujeme být o dané skutečnosti trvale informováni.

Další možnosti akustické signalizace představují telemetrické alarmy v nabídce *Timers/Sensors*→*Alarms* (Obrázek 27 vlevo). Zde je možné vzít libovolný údaj z čidel a k němu přiřadit příslušný audio soubor, který se přehraje po překročení určité přednastavené hodnoty. Volitelně může také dojít k odstavení motoru, jelikož většina dostupných alarmů souvisí s pohonným systémem a napětím baterií.



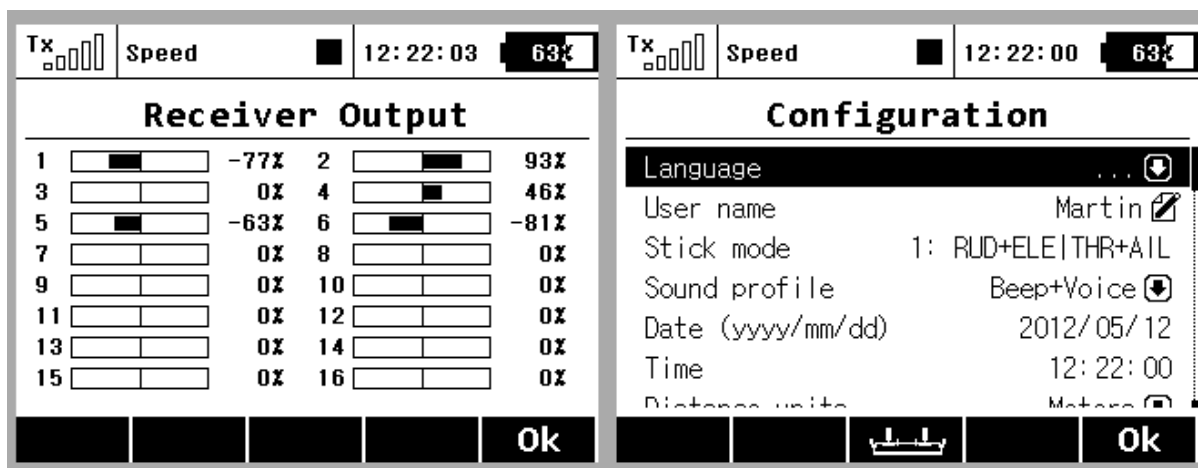
Obrázek 27: Nastavení alarmu na základě telemetrie a vedle detail časovače.

Ve vysílači je možné vytvořit až deset nezávislých časovačů. Ty lze používat velmi univerzálně, například k měření celkové doby letu nebo uplynulého motorového času. Formulář pro editaci jednoho časovače se nachází na Obrázku 27 a přistoupíme k němu přes sekvenci *Timers/Sensors→Timers→Add*. V tomto formuláři lze zvolit počáteční a koncový čas, tedy zda časovač bude fungovat v režimu odpočet nebo jako klasické inkrementální stopky. Dále je v závislosti na způsobu použití zvolen typ časovače a aktivační spínač. *Standardní* časovač běží právě tehdy, když je spínač v pozici *Zapnuto*; v každém jiném momentu je zastaven. Typ *Free Running*, tedy volně běžící časovač, je jednou spuštěn a běží neustále až do ukončení letu. S jeho pomocí je možné zajistit měření celkového letového času. Poslední dostupný typem má označení *Laps*. Slouží především jako pomůcka pro závodní piloty, neboť se zde měří i jednotlivé intervaly mezi stisky aktivačního spínače. Těchto mezičasů může být několik desítek a po skončení letu je možné si je prohlédnout.

Ve formuláři detailu časovače se ještě nachází položka *Report Type*. Ta umožňuje vybrat si ze tří přednastavených schémat akustického oznamování zbývajících času (*Beep 1*, *Beep 2 a Voice*). Jednotlivá schémata odpovídají příslušným souborům JSON ve složce **/Config** na SD kartě. Ty jsou přístupné k editaci, takže je uživateli dovoleno, aby si v textovém editoru vytvořil libovolné oznamovací schéma. Formát dat je jednoduchý, jak je vidět z následujícího úryvku kódu.

```
[
  {"Time":-3,"Type":1,"Freq":4000,"Cnt":1,"Length":200},
  {"Time":-2,"Type":1,"Freq":4000,"Cnt":1,"Length":200},
  {"Time":-1,"Type":1,"Freq":4000,"Cnt":1,"Length":200},
  {"Time":0,"Type":1,"Freq":4000,"Cnt":1,"Length":1000},
  {"Time":60,"Type":2,"File":"Audio.wav"}
]
```

Zde dojde v čase 60s od spuštění časovače k přehrání audio souboru s cestou **/Audio/Audio.wav**, dále jsou poslední tři sekundy před koncovým časem indikovány pípnutím. To, jestli se daná akce má provést v přesném čase po spuštění anebo před koncem odpočtu, je rozlišeno pomocí kladné, resp. záporné hodnoty položky *Time*.



Obrázek 28: Vlevo vizualizace výstupů přijímače, vpravo základní nastavení systému.

Poslední formuláře, které jsou v této práci popsány, se nacházejí v menu *System*. První formulář graficky znázorňuje výstupy všech kanálů (*Receiver Output*, Obrázek 28 vlevo), takže je možné ověřit správnost provedených změn v nastavení i bez nutnosti mít zapnutý model. Jestliže je model dobře nakonfigurován, výstupy jednotlivých kanálů se mění s tím, jak pilot hýbe pákami.

Formulář *Configuration* slouží k nastavení všech základních systémových proměnných. Jak je vidět na obrázku vpravo, jsou to většinou běžné údaje jako čas a datum, jméno vlastníka nebo zvolený jazyk. V tomto formuláři je dále možné změnit logický mód křížových ovladačů (řádek *Stick mode*, položky odpovídají Obrázku 6). Nakonec zde můžeme zvolit, v jakých jednotkách se bude na hlavní obrazovce ukazovat vzdálenost (metry nebo stopy) a teplota (stupně Celsia či Fahrenheita).

7 Shrnutí

Prvotní analýza softwarových požadavků byla provedena v březnu 2011. Jejím cílem bylo nastínit schéma hlavních kategorií v menu a posléze i navrhnout kompletní grafickou podobu jednotlivých nabídek. V pozdějších etapách vývoje byla specifikace upřesňována, případně přepracována dle připomínek zkušených modelářů, kdy musela být např. znovu vytvořena celá transformační funkce a jí odpovídající formuláře v kategorii *Fine Tuning*.

V průběhu vývoje také rostly nároky aplikace na výpočetní výkon a hlavně paměťový prostor (viz podkapitola Metriky). Z tohoto důvodu se postupně měnil typ použitého hlavního mikrokontroléru. Původně byl použit čip STM32F103RCT6, jenž obsahoval pouhých 48kB RAM a také se projevoval nedostatek paměti Flash. Tento mikrokontrolér byl následně nahrazen typem STM32F103ZGT6, který již v mnoha ohledech vyhovoval. Ovšem vzhledem k tomu, že životnost vysílače může být velmi dlouhá a v plánu je ještě mnoho rozšiřujících aplikací, pro budoucnost se tato alternativa ukázala jako omezující. Proto byl nakonec pro sériovou výrobu zvolen čip STM32F405RGT6, jehož parametry jsou uvedené v kapitole 5. Jelikož se stále jedná o konstantního výrobce, jenž se snaží být se svými produkty zpětně kompatibilní, byl přechod na novou řadu vždy relativně snadný.

Již v průběhu tvorby diplomové práce probíhala širší fáze testování, kdy vysílač obdržela celá řada profesionálních modelářů z Česka, Německa a Rakouska za účelem připomínkování. Ohlasy byly zatím veskrze pozitivní a lze říci, že vysílač JETI DC-16 splnil velká očekávání, která si dal za cíl. Distribuce bude probíhat společně i s kufrem a závěsným pultem, jak je to ilustrováno na Obrázku 29.



Obrázek 29: Hotový prototyp vysílače i s kufrem a závěsným pultem.

7.1 Metriky

V Tabulce 4 jsou shrnuty základní metriky výsledného systému. Jak můžeme vidět, aplikace je na poměry známé z běžných vestavěných systémů relativně rozsáhlá a zabírá mnoho cenných prostředků. Počet řádků byl měřen programem *Cloc* [20], nároky jednotlivých komponent na paměť RAM byly zjišťovány pomocí funkce *mallinfo*. Paměť RAM je využita z necelých šedesáti procent.

Nároky na kód a firmware	
Počet řádků kódu celkem i s ext. knihovnami	140 000
Počet řádků kódu vlastní aplikace	47 000
Celkový počet formulářových oken	73
Velikost binárního souboru bootloaderu	22kB
Velikost binárního souboru hlavní aplikace	395kB
Nároky na paměť RAM	
Audio subsystém	cca 40kB v závislosti na vzorkovací frekvenci
Grafický subsystém	10kB buffer + 5-30kB formuláře
Lokalizované řetězce	cca 15kB podle zvoleného jazyka
Konfigurace modelu	10kB – 30kB
Doba odezvy na změnu řídicího vstupu	Max. 25ms
Použité vývojové prostředí	Ride7 (Raisonance), Qt (Nokia)
Kompilátor	GCC 4.6 pro x86 a ARM

Tabulka 4: Metriky kódu vytvořeného software a jeho nároky na prostředky.

Doba odezvy na vstup byla měřena s využitím speciálně zkonstruovaných stopek, a to tak, že se měřil čas od okamžiku přepnutí spínače do adekvátní reakce na výstupu přijímače. Na tuto dobu má největší vliv interval mezi jednotlivými iteracemi při přeposílání paketů, který činí rovných 20ms. Interní zpracování dat ve vysílači trvá maximálně 3ms a další 2ms zabere přeposílání paketu bezdrátovou linkou. Latence je tedy vyhovující, neboť je dosaženo plynulého a rychlého řízení modelu.

8 Závěr

Cílem diplomové práce bylo nastudování problematiky dálkového řízení modelů, seznámení se s hlavními představiteli na poli profesionálních RC souprav a na základě nabytých znalostí vytvoření konkurenceschopného softwarového produktu propojeného s konkrétní hardwarovou platformou.

V této práci byly prezentovány základní poznatky z modelářské terminologie a následně byly formulovány požadavky na nový vysílač. Podle těchto požadavků bylo nakonec navrženo a implementováno grafické uživatelské rozhraní spolu s hlavním řídicím programem vysílače. Funkčnost navrženého ovládacího programu byla úspěšně ověřena během prvních zkušebních letů zkušebních letů.

Testovací piloti z řad profesionálů, kteří obdrželi první prototypové kusy vysílače, již teď slaví úspěchy. Tím zatím nejlepším je první místo Rakušana Gernota Bruckmanna v halové akrobacii na Indoor World Championship - ETOC 2012, který se pořádá v americkém Toledu.

Praktická část této práce byla rovněž úspěšně obhájena před komisí EEICT 2012 v kategorii Elektronika a komunikace a mezi ostatními pracemi se umístila na třetím místě.

Vzhledem k možnostem použitého mikrokontroléru je již v současné době zřejmé, že uvedením produktu na trh vývoj nekončí. Nabízí se několik funkcí, které by bylo nyní vhodné dále naimplementovat. Jedná se např. o akustickou signalizaci varia (tj. intenzity stoupání nebo klesání), akustické hlasové oznamování aktuálních hodnot telemetrie nebo online bezdrátové připojení k PC za účelem vyhodnocování telemetrických dat během letu.

9 Použité zkratky a termíny

Modelářské servo – miniaturizovaný elektromechanický obvod, který slouží k řízení pohyblivých ploch modelu a pomocí zpětnovazebního členu dokáže udržet přesnou polohu.

RC – Remote Control/Radio Controlled. Dálkově ovládaný, rádiem řízený.

Rx – Receiver – označení pro přijímač umístěný v modelu.

Tx – Transmitter – označení pro vysílač (nebo jen vysílací modul), který je umístěn v ruce pilota.

DR – Dual Rate – dvojí výchylky. Důležitá vlastnost u vysílače pro modely, které jsou určeny jak pro akrobacii, tak pro rekreační let. Pro akrobacii se obvykle nastavují maximální výchylky na kormidlech, takže má model velmi ostré a razantní reakce. Naopak při rekreačním letu jsou nastaveny výchylky menší, neboť jsou požadovány také přiměřeně mírné odezvy modelu.

Expo – Exponenciální průběh dráhy serva – umožňuje nelineární odezvu výstupu serva na vstup ovladače. Obvykle s ohledem na letové vlastnosti modelu je možné plynule nastavit v blízkosti neutrální polohy ovladače požadovanou velikost odezvy serva.

Režim Učitel/Žák – Obvykle propojení dvou vysílačů takovým způsobem, že ten hlavní (učitel) primárně ovládá model a je schopen ve vhodném okamžiku předat řízení podřízenému vysílači (žák). Využívá se především při výcviku mladých pilotů.

Trim – Prvek sekundárního řízení, jímž se jemně docílí vyrovnaného přímočarého letu.

Subtrim – Posun celé dráhy serva o daný ofset. Středová pozice serva je změněna a velikosti jeho výchylek jsou zachovány vzhledem k této pozici.

FHSS – Frequency Hopping Spread Spectrum – technologie použitá pro digitální přenos dat v nízkém bitovém toku, ale s vysokou spolehlivostí. Mnohokrát za sekundu dochází k vzájemnému přeladování vysílače a přijímače mezi všemi frekvencemi pásma, takže je obtížné rušení i odposlech takové komunikace.

Fail-Safe – Bezpečnostní poloha, kterou je možné nastavit pro libovolné servo a do níž je uvedeno po výpadku signálu.

Proporcionální funkce – výchylka serva je úměrná výchylce příslušného ovladače.

Neporcionální funkce – výchylka je nastavena pouze dvěma krajními polohami.

UART, USART – Universal (Synchronous) Asynchronous Receiver/Transmitter – základní sériová sběrnice používaná v komunikaci mezi komponentami embedded systémů. Pokud je asynchronní, data jsou odvozena od domluveného taktu. Jestliže je provedena jako synchronní, pak je u datových vodičů navíc vodič se zdrojem taktu.

SPI – Serial Peripheral Interface – synchronní sběrnice. Aby byla dosažena správná funkčnost, musí být jeden obvod označen jako Master a druhý jako Slave.

Literatura

- [1] JEŽERSKÝ, J.: *Serva a jejich ovládání*. [online].
[cit. 9. 1. 2012]. <<http://www.serva.cz/rizeni-serva-teorie/>>
- [2] DEPARTMENT OF THE NAVY: *Naval Aviation Chronology in World War II*. Naval Historical Center. Retrieved 2007-04-10.
- [3] ZVĚDĚLÍK, Z.: *Základní pojmy z RC světa*. Listopad 2004. [online].
[cit. 9. 1. 2012]. <<http://www.mojehobby.cz/magazin/detail.php?idclanku=15>>
- [4] *Jak funguje modelářské servo*. [online]. <<http://vlastikd.webz.cz/bastl/serva.htm>>
- [5] JANKO, O.: *Výhody digitálních serv voči servám analogovým – 1.část*. [online] [cit. 9. 1. 2012]. <<http://rcmodely.cevaro.sk/index.php?id=245&n=vyhody-digitalnych-serv-1cast>>
- [6] CoCoX CoOS. *Free and open real-time Operating System* [online].
[cit. 26. 3. 2012]. <<http://www.coocox.org/CoOS.htm>>
- [7] FatFs Generic FAT File System Module. [online].
[cit. 9. 1. 2012]. <Online http://elm-chan.org/fsw/ff/00index_e.html>
- [8] RM0008 Reference manual. *STM32F101xx, STM32F102xx, STM32F103xx, STM32F105xx and STM32F107xx advanced ARM-based 32-bit MCUs*. [online]. <<http://www.st.com>>.
- [9] Kolektiv autorů: *Návod k obsluze FF-10 (T-10CP)*. Robbe Modellsport GmbH & Co.KG.
- [10] Kolektiv autorů: *Graupner MC-32 Programmier-Handbuch*. Graupner GmbH & CO. KG.
- [11] Kolektiv autorů: *T18MZ, 18-Channel Computer Systems, Instruction manual*. Futaba Corporation, 2011.
- [12] HANÁČEK, P.: BMS 0x02. *Přístup k médiu. Presentace k předmětu Bezdrátové a mobilní sítě*. Zimní semestr 2011.
- [13] Kolektiv autorů: *CIAM Volume F5 Radio Control Electric Powered Model Aircraft*. Fédération Aéronautique Internationale. [online].
[cit. 20. 5. 2012]. < <http://www.fai.org/ciam-our-sport/f5-electric-flight>>
- [14] LNĚNIČKA, J.: *O dřívějších větroních a modelech*. Hradec Králové, Aeromodel, 1996.
- [15] LNĚNIČKA, J.: *Letecké modelářství a letectví*. Hradec Králové, Aeromodel, 1996. 208 s.
- [16] Jetimodel, s. r. o.: Vysílače DC-16 a DS-16 [online]. Jetimodel Příbor.
[cit. 26. 4. 2012]. <<http://www.jetimodel.cz/?page=article&id=25>>.
- [17] DROZDEK, A.: *Data Structures And Algorithms In C++*, 3. vydání. Cengage Learning, 2005. ISBN 9780534491826. 758 s.
- [18] CROCKFORD, D.: *RFC 4627, JSON.org*. July 2006. [online].
[cit. 20. 5. 2012]. < <http://www.ietf.org/rfc/rfc4627.txt?number=4627>>
- [19] VALIN, J. M.: *The Speex Codec Manual, Version 1.2 Beta 3* [online].
[cit. 19. 5. 2012]. <<http://www.speex.org>>
- [20] CLOC: *Count Lines of Code* [online]. 2012 [cit. 20. 5. 2012]. <<http://cloc.sourceforge.net/>>

Dodatek A – Nastavení vývojového prostředí a ovládání aplikace emulátoru

Emulátor softwaru vysílače Jeti DC-16 je přiložen na doprovodném médiu. Byl vytvořen především pro urychlení vývoje a snadnou testovatelnost. Tento program dokáže z větší části napodobit skutečný vysílač. Všechny uživatelské nabídky jsou v něm dostupné, pouze je omezena funkčnost přístupu k souborovému systému a operace týkající se emulování jiných hardwarových komponent nebyly implementovány vůbec.

Aplikace byla implementována v prostředí Qt 4.7.4 pro Windows a byly pro ni z větší části použité stejné zdrojové soubory jako pro fyzické zařízení. Na doprovodném CD je přiložen zkompilevaný program i soubor projektu. Ten pak stačí otevřít v Qt Creatoru a přeložit. Aplikace se spouští standardně bez jakýchkoli parametrů.

Typický vzhled emulátoru je na Obrázku 30. Ovládání je možné buď myší klikáním na tlačítka formuláře, nebo pomocí klávesnice. Následující seznam shrnuje význam použitých kláves.

[Šipka nahoru] – pohyb rotačním enkodérem ve směru hodinových ručiček.

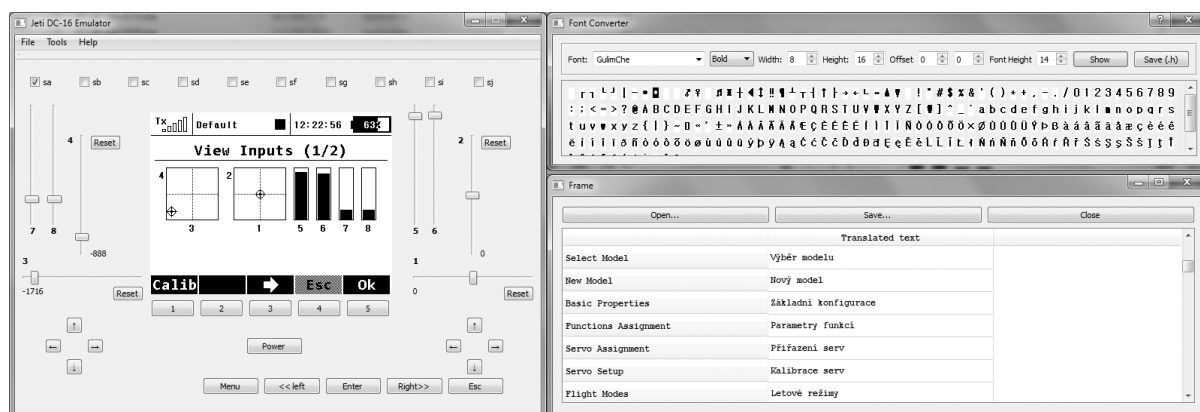
[Šipka dolů] – pohyb enkodéru proti směru hodinových ručiček.

[Šipka doprava] – stisk enkodéru, potvrzení volby.

[Šipka doleva], [ESC] – stornování volby.

[Mezerník] – zmáčknutí tlačítka Menu.

Klávesy [1] až [5] – stisk odpovídajícího tlačítka pod displejem.



Obrázek 30: Emulátor vysílače pro PC. Obsahuje doplňkové funkce pro jazykovou lokalizaci řetězců a konvertování fontů do vhodného formátu pro kompilaci.

Dodatek B – Struktura doprovodného CD

`/Readme.txt` – text s popisem obsahu CD

`/bin/` – adresář s předkompilovanou binární podobou počítačového emulátoru. Obsahuje i potřebné knihovny.

`/report/` – složka s kompletní technickou zprávou diplomové práce ve verzi pro tisk.

`/src/` – adresář se zdrojovými kódy emulátoru a výsledné aplikace.

`/video/` – dvě ukázková videa s testem funkčnosti časovačů a akustického hlášení.